

AD-A126 980

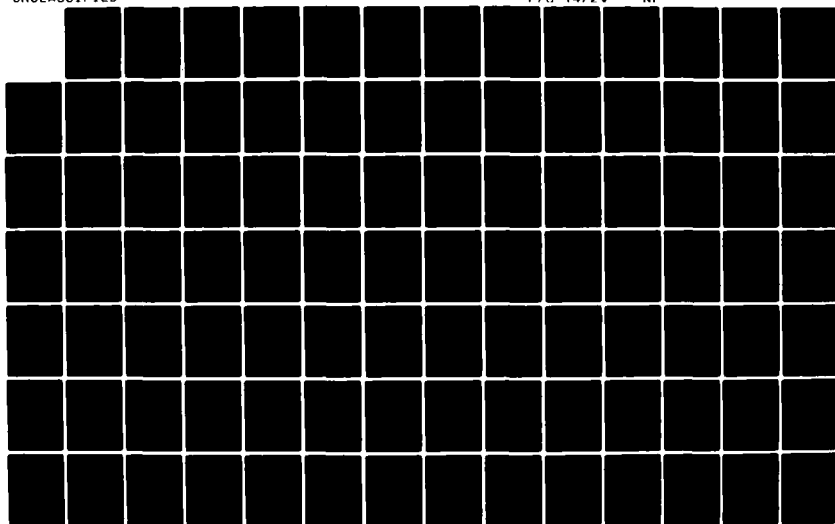
A MICROCOMPUTER-BASED SAMPLING DIGITAL VOLTMETER(U)
NAVAL RESEARCH LAB WASHINGTON DC R E SCOTT ET AL.
31 MAR 83 NRL-MR-4872

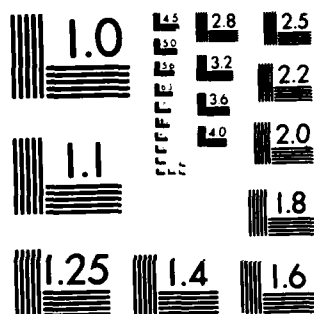
1/2

UNCLASSIFIED

F/G 14/2.

NI





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ADA 126980

2
NRL Memorandum Report 4872

A Microcomputer-Based Sampling Digital Voltmeter

RICHARD E. SCOTT, JR. AND JAMES D. GEORGE

*Acoustical Systems Branch
Underwater Sound Reference Detachment
Naval Research Laboratory
P.O. Box 8337
Orlando, Florida 32856*

March 31, 1983



NAVAL RESEARCH LABORATORY
Washington, D.C.

Approved for public release; distribution unlimited.

DTIC FILE COPY

83 04 19 069

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM												
1. REPORT NUMBER NRL Memorandum Report 4872	2. GOVT ACCESSION NO. AD-A126 880	3. RECIPIENT'S CATALOG NUMBER												
4. TITLE (and Subtitle) A Microcomputer-Based Sampling Digital Voltmeter		5. TYPE OF REPORT & PERIOD COVERED Interim report on a continuing problem												
		6. PERFORMING ORG. REPORT NUMBER												
7. AUTHOR(s) Richard E. Scott, Jr. and James D. George		8. CONTRACT OR GRANT NUMBER(s)												
9. PERFORMING ORGANIZATION NAME AND ADDRESS Underwater Sound Reference Detachment Naval Research Laboratory P.O. Box 8337, Orlando, FL 32856		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Element: 62711N Project NRL Work Unit: (59)-0593												
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Sea Systems Command (NAVSEA 63R12) Washington, DC 20362		12. REPORT DATE 31 March 1983												
		13. NUMBER OF PAGES 109												
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED												
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A												
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.														
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)														
18. SUPPLEMENTARY NOTES This work was performed in the Acoustic Metrology Program sponsored by the Naval Sea Systems Command (SEA63R12).														
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) <table border="0"> <tr> <td>Vector voltmeter</td> <td>Distortion measurement</td> <td>Kaiser window</td> </tr> <tr> <td>True rms</td> <td>A/D converter</td> <td>Microcomputer</td> </tr> <tr> <td>Phase measurement</td> <td>SFDFT</td> <td>PDP-11</td> </tr> <tr> <td></td> <td></td> <td>IEEE-488 bus</td> </tr> </table>			Vector voltmeter	Distortion measurement	Kaiser window	True rms	A/D converter	Microcomputer	Phase measurement	SFDFT	PDP-11			IEEE-488 bus
Vector voltmeter	Distortion measurement	Kaiser window												
True rms	A/D converter	Microcomputer												
Phase measurement	SFDFT	PDP-11												
		IEEE-488 bus												
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) <p>This report describes a sampling digital voltmeter developed at the Underwater Sound Reference Detachment of the Naval Research Laboratory. It consists of a PDP-11/23 microcomputer with up to three channels of analog-to-digital converters capable of making simultaneous measurements on pulsed or continuous sinusoids at frequencies up to 100 kHz. The voltmeter is controlled by ASCII commands on the IEEE-488 bus. Statistical parameters (mean, peak, standard deviation, and second joint moment) are computed using numerical integration. Sinusoidal parameters (amplitude, phase, and harmonic (continued)</p>														

DD FORM 1473
1 JAN 73EDITION OF 1 NOV 65 IS OBSOLETE
5/N 0102-LF-014-6601

1

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(Item 20 continued)

distortion) are computed using single-frequency discrete Fourier transforms (SFDFT). The digital filtering resulting from the use of the SFDFT can be further enhanced by optional data sequence weighting (multiplying the sequence by a given windowing sequency). Computation times may be reduced by optional subsequence averaging. Complete assembly language listings for the voltmeter program are included and described.

CONTENTS

INTRODUCTION.....	1
APPLICATIONS.....	1
THEORY OF OPERATION.....	2
COMPUTATION PROCEDURES.....	4
Sampling Criteria.....	4
Sinusoidal Parameters.....	6
Digital Filtering.....	7
Statistical Parameters.....	10
SOFTWARE FUNCTIONS.....	11
SUMMARY	13
ACKNOWLEDGMENTS.....	14
REFERENCES.....	14
APPENDIX A - Voltmeter Hardware.....	15
APPENDIX B - Voltmeter Commands.....	17
APPENDIX C - Voltmeter Status Messages.....	21
APPENDIX D - Voltmeter Control Subroutines.....	23
APPENDIX E - Voltmeter Program Listings.....	31
APPENDIX F - Voltmeter Memory Map.....	109



A

A MICROCOMPUTER-BASED SAMPLING DIGITAL VOLTMETER

INTRODUCTION

Acoustical measurement systems at the Underwater Sound Reference Detachment (USRD) of the Naval Research Laboratory need a specialized multiple-channel voltmeter with the capability to measure voltage, relative phase between channels, and harmonic distortion for pulsed sinusoids at frequencies between 0.1 Hz and 100.0 kHz. To answer this need, a sampling digital voltmeter was developed around an LSI-11/23 microcomputer, using three analog-to-digital (A/D) converters and an IEEE-488 interface to serve as the communications link with any appropriate host computer. Using this voltmeter, a "simple" computer-controlled measurement system can then be configured with the following auxiliary equipment as shown in Fig. 1: synthesizers to generate a sinusoidal test signal and a coherent sampling signal, and programmable amplifiers or attenuators as needed. (The voltmeter has no intrinsic autoranging capability.)

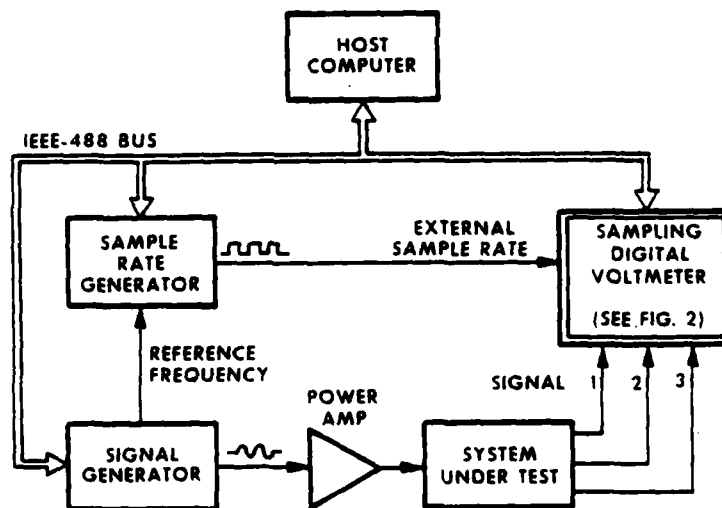


Fig. 1 - Typical measurement system using the microcomputer-based sampling voltmeter.

APPLICATIONS

The two immediate applications for such a voltmeter were for the USRD's Anechoic Tank Facility (ATF) and Low-Frequency Facility (LFF) measurement systems.

The ATF high-power measurement system requires a three-channel voltmeter to measure simultaneous projector drive voltage, drive current, and receiver output (hydrophone voltage). Measurements of relative phase between voltage and current and the harmonic distortion in all three channels are necessary. The practical frequency range is 2 to 100 kHz; the mode of operation is usually pulsed.

The LFF system requires a two-channel voltmeter to measure the simultaneous voltage outputs of two hydrophones, generally a probe standard and an unknown hydrophone. Relative acoustic phase is required, and harmonic distortion is a useful parameter to monitor. The practical frequency range is 0.1 Hz to 4 kHz; the mode of operation is continuous wave (cw).

The voltmeter was, therefore, conceived to be a device independent of (and transportable between) specific measurement systems at the USRD, with the possible exception being that the A/D converters might vary with application (e.g., a need to operate at higher sampling rates) and are otherwise functionally identical plug-in modules. To get the widest possible range of potential host computers, communication through the IEEE-488 bus was chosen. Another requirement was that all external hardware, including the sample-rate generator, be controlled by the host computer for maximum flexibility. Therefore, all external functions, including automatic ranging of the voltmeter, are controlled by the host.

Since initiating and implementing this signal processing sampling voltmeter, two other similar instruments have been described or released: the low-frequency sampling voltmeter by the National Bureau of Standards [1] and the sampling network analyzer by the Dranetz Company [2]. The three instruments share the sampling ideas but differ in the features and algorithms implemented.

THEORY OF OPERATION

The voltmeter consists of a microcomputer containing up to three A/D converters with direct memory access (DMA) capability (Fig. 2). This configuration allows the computer to digitize up to three input waveforms simultaneously, sampling them when triggered by a common external sample-rate generator. Where simultaneous measurement of all channels is not desired, separate or delayed sampling signals can be given to individual A/D converters. The present voltmeter is designed to digitize three waveforms of up to 1024 samples each; these buffers could be expanded to 3900 samples (see Appendix F).

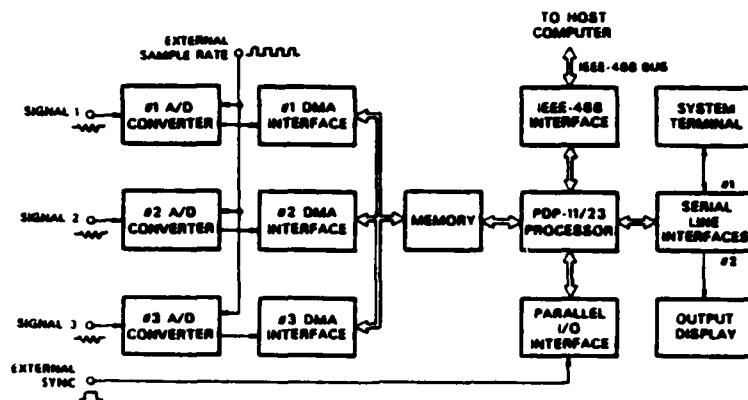


Fig. 2 - Block diagram of the microprocessor-based sampling voltmeter.

The DMA interface loads 12-bit digitized values directly into memory at rates up to 100 kHz per channel. The individual sampled waveforms can be transmitted to the host for processing; alternately, the slave can compute sinusoidal and statistical parameters from these waveforms and return them to the host. Parameters are computed for the fundamental frequency. (Throughout this report "fundamental" frequency refers to the sinusoidal test frequency being measured, no matter how many cycles of this frequency are included in a measured sequence.)

The sinusoidal quantities amplitude, phase, and harmonic distortion are computed for each channel using single frequency discrete Fourier transforms (SFDFT). The statistical quantities average, standard deviation, and the positive and negative extrema are obtained for each channel to characterize noisy or nonsinusoidal signals. In voltmeter terms, these statistical parameters are the dc, true ac rms, and positive and negative peak voltages. To measure the coherence or correlation between channels, the second order joint moment statistic, $E(xy)$, can be computed. For the case of the two signals, x and y ; proportional to current and voltage, the joint moment is proportional to power. The covariance and cross correlation can be readily obtained from these parameters.

COMPUTATION PROCEDURES

Sampling Criteria

In the usual case of interest (pulsed sinusoids), sampling errors are minimized through coherent sampling, which requires an integer number of both test frequency cycles and sampling periods:

$$\frac{m}{f_T} = \frac{n}{f_S} \quad (1)$$

where f_T is test frequency
 f_S is sampling frequency
 n is samples per sequence (integer)
 m is cycles per sequence (integer).

If the sampling frequency,

$$f_S = \frac{n}{m} f_T \quad (2)$$

is greater than twice the test frequency, the Nyquist theorem is satisfied, and satisfactory results will be obtained. If the necessary sampling rate is too fast for the A/D converter to operate properly, undersampling can be used; by choosing fewer than two samples per cycle while retaining an integer number of samples and cycles per sequence, an effective higher sampling rate is attained. A comparison of the oversampled waveform shown in Fig. 3 with the undersampled waveform in Fig. 4 clearly demonstrates how undersampling can synthesize a waveform (Fig. 5) effectively sampled at a higher rate, although with fewer cycles overall. Undersampling will only work properly on periodic waveforms, however. Computed data in Table 1 demonstrates that peak voltage and harmonic distortion may be affected slightly by undersampling.

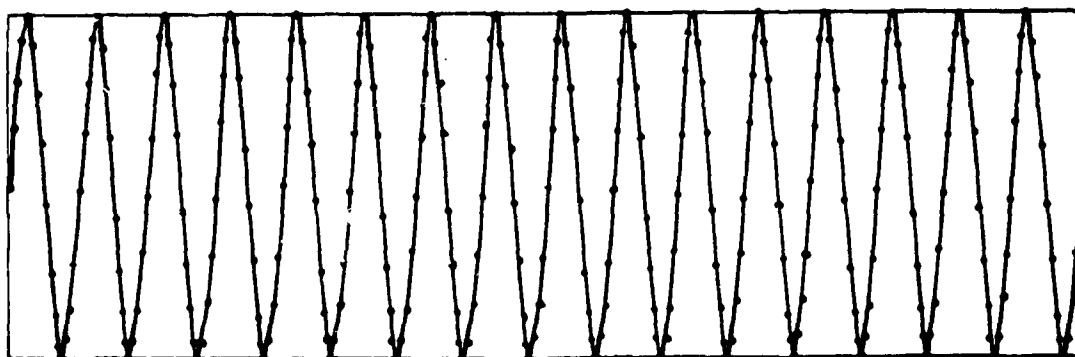


Fig. 3 - Steady-state sinusoidal waveform sampled for 16 cycles at 17 samples per cycle: 272 samples (oversampling).

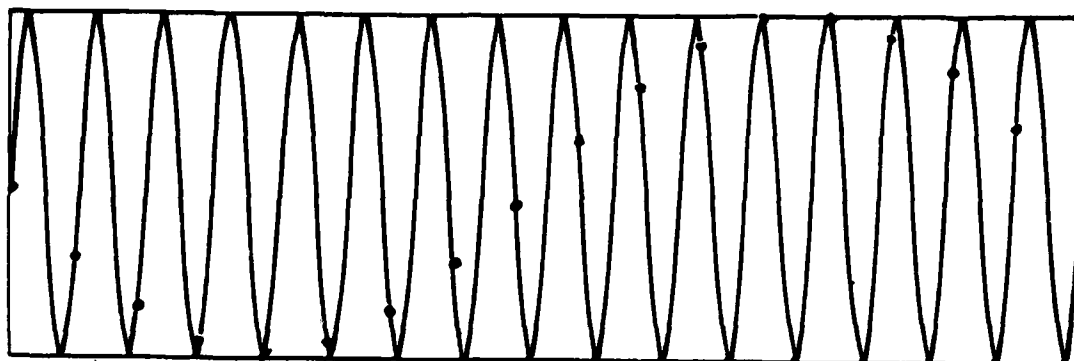


Fig. 4 - Steady-state sinusoidal waveform sampled for 16 cycles at 1.0625 samples per cycle: 17 samples (undersampling).

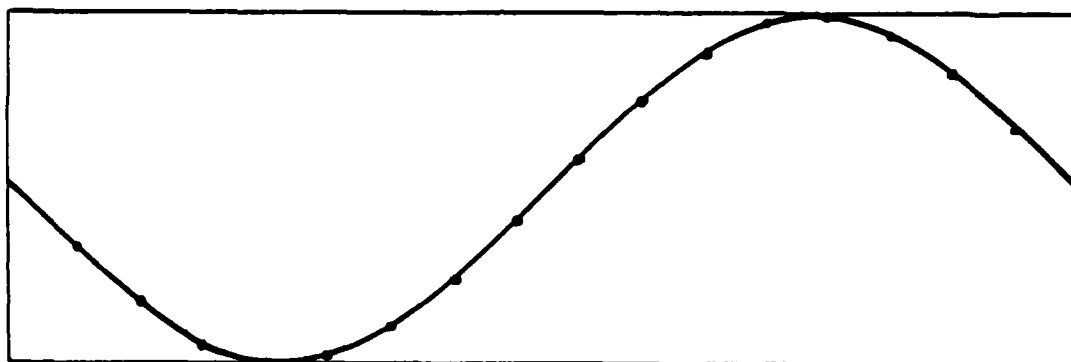


Fig. 5 - Steady-state sinusoidal waveform synthesized from the 17 samples shown in Fig. 5.

Table 1 - Comparison of oversampled and undersampled measurements.

COMPUTED PARAMETER	OVER SAMPLING	UNDER SAMPLING
RMS VOLTAGE (STD DEV)	1.047	1.047
PEAK VOLTAGE	1.485	1.492
DC VOLTAGE (MEAN)	-0.013	-0.013
TOTAL POWER	1.102	1.102
RMS VOLTAGE (DFT)	1.047	1.047
RELATIVE PHASE	0	0
HARMONIC DISTORTION	0.100	0.095

Normal: 2 cycles at 26 samples per cycle (52 samples total).

Undersampled: 25 cycles at 1.04 samples per cycle (52 samples total).

Sinusoidal Parameters

For sinusoidal testing, the signals are of the form

$$y(t) = \sqrt{2}A_m \cos(2\pi f_T t + \phi_m) + \sum_{k=1}^{\ell} \sqrt{2}A_{km} \cos(2\pi k f_T t + \phi_{km}) + n(t) \quad (3)$$

where the terms are, respectively, the fundamental, the harmonics, and the additive noise.

Least squares estimates of the rms amplitude, A_m , and phase, ϕ_m , of the fundamental are obtained from

$$\hat{A}_m = \sqrt{2} [\text{Re}^2(Y_m) + \text{Im}^2(Y_m)]^{1/2} \quad (4)$$

and

$$\hat{\phi}_m = \tan^{-1} [\text{Im}(Y_m)/\text{Re}(Y_m)] \quad (5)$$

where

$$Y_m = \frac{1}{n} \sum_{i=0}^{n-1} y_i \exp(-j2\pi im/n) \quad (6)$$

which is the m th component of the DFT of the n -point sequence y_i containing m cycles of the fundamental; that is, the single frequency discrete Fourier transform (SFDFT). With coherent sampling and in the absence of noise, A_m and ϕ_m are exact.

Harmonic distortion, D , is defined in terms of the ratio of power in the harmonics to power in the fundamental.

$$D = \left[\sum_{k=2}^{\ell} A_{km}^2 / A_m^2 \right]^{1/2} \quad (7)$$

Since the sequences are relatively short and few lines are required, Goertzel's algorithm [3,4] is an efficient implementation of the SFDFT.

Digital Filtering

The evaluation of the DFT at a single frequency, as in Eq. (6) can be interpreted as a digital filter [5] with a corresponding enhanced signal-to-noise ratio. However, the filter quality, the skirts and side lobes, can be significantly improved by weighting or windowing the data sequence and then doing the SFDFT. A particularly effective choice of data weighting sequence is the Kaiser I_0 -sinh window [6]. An example of a weighted data sequence is shown in Fig. 6, and the corresponding digital filter responses are shown in Fig. 7.

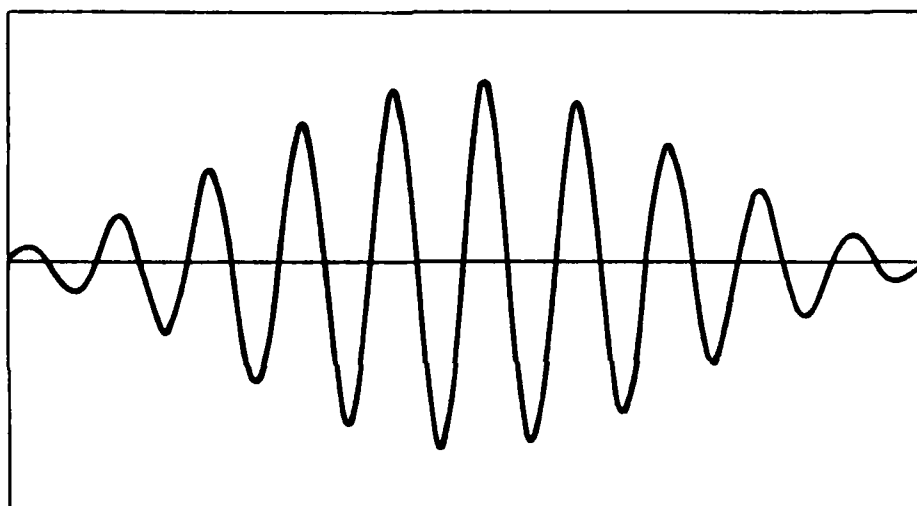


Fig. 6 - Sinusoidal waveform of 10 cycles (product with Kaiser window with 40 dB of spectral side lobe attenuation).

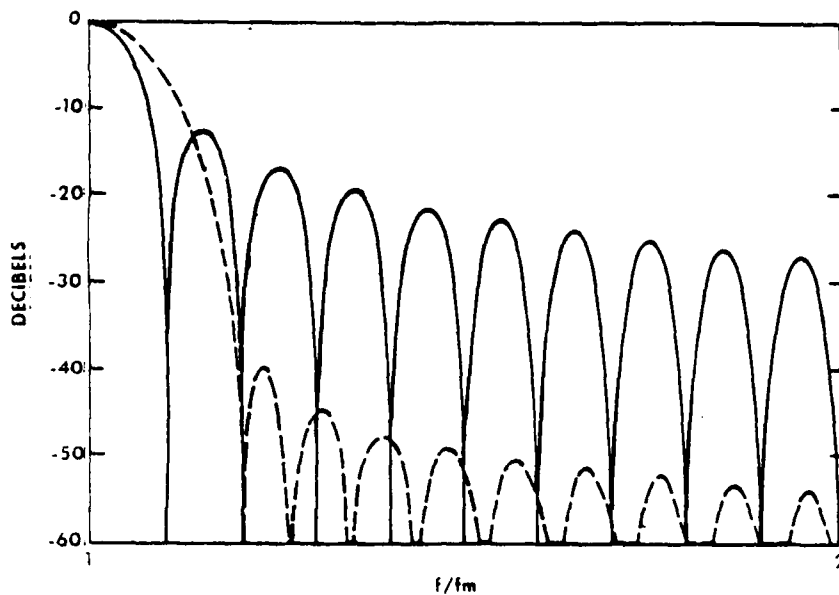


Fig. 7 - Comparison of the filter effect of an unweighted SFDF (solid line) with that of a SFDF with the Kaiser window with 40 dB of side lobe attenuation (dashed line). The frequency axis is normalized to the center frequency f_m which corresponds to the m th spectral line. The data sequence includes 10 cycles.

The filters are centered on the m th spectral line or test frequency, f_T . Following convention, the filter bandwidth is defined as the main lobe width or the distance between the minima adjacent to the m th line. For the unweighted SFDF, the fractional bandwidth or main lobe width is

$$\left(\frac{\Delta f}{f_m} \right)_{\text{Rectangular}} = \frac{2}{m}, \quad (8)$$

where m is the number of cycles of f_T in the data sequence.

The reduced side lobes of the Kaiser filter are achieved at the expense of an increased main lobe width

$$\left(\frac{\Delta f}{f_m} \right)_{\text{Kaiser}} = \left(\frac{2}{m} \right) \frac{6(R + 12)}{155}, \quad (9)$$

where R is the attenuation in dB of the first side lobe. Figure 8 illustrates the increase in filter main lobe width as the side lobes are reduced.

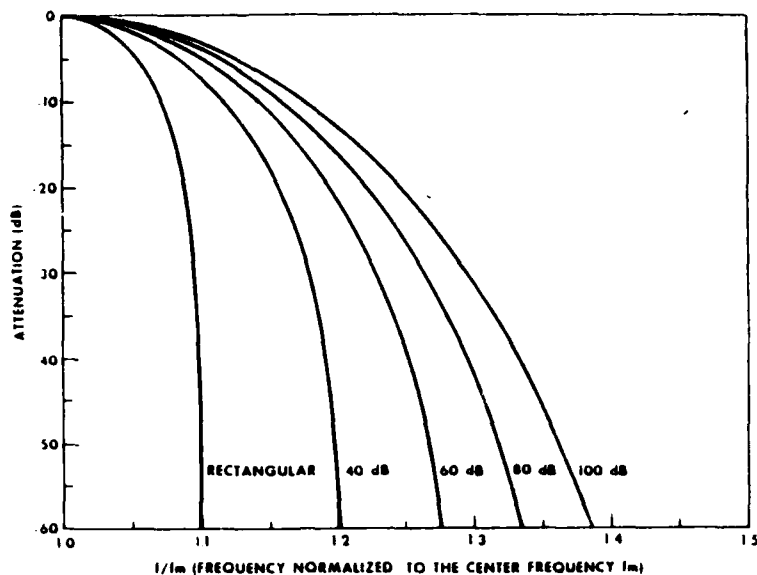


Fig. 8 - Comparison of filter main lobes for unweighted SFDF and Kaiser-weighted SFDF with 40 to 100 dB side lobes. The side lobes are not plotted for clarity. The data sequence includes 10 cycles ($m = 10$).

The data window or weighting sequence, w_i , is computed in the host and transferred to the voltmeter where the window is applied in estimating amplitude and phase using the SFDF of the weighted sequence

$$Y_m = \frac{1}{n} \sum_{i=0}^{n-1} w_i y_i \exp(-j2\pi im/n) \quad (10)$$

The amplitude estimate, A_m , is corrected by the window scale factor

$$\bar{w} = \frac{1}{n} \sum_{i=0}^{n-1} w_i \quad (11)$$

Table 2 illustrates that the Kaiser window does not alter or bias the amplitude and phase estimates at the test frequency. However, the harmonic distortion, particularly for low distortion, is biased by the window; larger sidelobe attenuations reduce this bias by reducing the amplitude of the harmonics created by the window.

Table 2 - Amplitude, phase and harmonic distortion estimates with and without a Kaiser window.

WINDOW	MEASURED DATA			SIMULATED DATA		
	\hat{A}_m	$\hat{\phi}_m$	100D	100D		
	RMS AMPLITUDE	PHASE DEGREES	PERCENT HARMONIC DISTORTION	PERCENT HARMONIC DISTORTION		
				0%	0.1%	1%
No Window	1.054	-123.3	0.060	0.000	0.100	1.000
- 30 dB Side Lobe	1.054	-123.3	0.071	0.032	0.073	0.972
- 40 dB Side Lobe	1.054	-123.3	0.077	0.018	0.084	0.984
- 50 dB Side Lobe	1.054	-123.3	0.082	0.009	0.092	0.992
- 60 dB Side Lobe	1.054	-123.3	0.087	0.004	0.097	0.997
- 70 dB Side Lobe	1.054	-123.3	0.097	0.002	0.099	0.999
- 80 dB Side Lobe	1.054	-123.3	0.093	0.001	0.099	0.999
- 90 dB Side Lobe	1.054	-123.3	0.095	0.000	0.100	1.000
-100 dB Side Lobe	1.054	-123.3	0.097	0.000	0.100	1.000

Computations performed on 20 cycles of data at 32 points per cycle. Measured data was the output of a synthesizer; simulated data was a computed sine wave with distortion added to the second harmonic as labeled.

When long sequences or long integration times are used to combat noise, the computation time can be significantly reduced by averaging. This efficiency is possible if the total sequence consists of two or more subsequences with an integer number of samples and cycles per subsequence. The simplest example is a data sequence of m cycles of the fundamental test sinusoid with each cycle containing n samples--this data sequence can be compressed from m cycles to one cycle by averaging. Both the SFDF and windowed SFDF computations can be speeded up through averaging. (The window is applied before averaging.)

Statistical Parameters

For signals which may be arbitrary periodic waveforms or noise-like data, statistical parameters are more appropriate in characterizing the data than are the SFDF derived amplitude and phase (although naturally these parameters can be computed for sinusoidal signals as well).

The voltage extremes or range of the waveform are obtained by inspection of the data sequence, and both the maximum and minimum voltages are found.

The average or dc voltage is computed with

$$y_{dc} = \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i \quad (12)$$

The standard deviation or true ac rms is obtained from

$$y_{rms} = \sigma_y = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2} \quad (13)$$

The choice of factor $1/n$ instead of $1/(n-1)$ is to satisfy the Parseval relationship so that the rms amplitude of an ideal sine wave is the same whether estimated with the SFDFIT or with the true rms methods.

The second joint moment

$$E(xy) = \frac{1}{n} \sum_{i=1}^n x_i y_i, \quad (14)$$

is analogous to power if the data sequences x_i and y_i are proportional to current and voltage. Power measurement was the motivation for including this joint moment estimate. The total ac power, P_{ac} , is computed using

$$P_{ac} = E(xy) - \bar{x} \bar{y}. \quad (15)$$

All of the summations in Eqs. (10) through (15) can be interpreted as Euler-Maclaurin approximations to integrals of the continuous time functions. For functions that can be represented as a finite number of harmonically-related sinusoids, this simple approximation is exact [7].

SOFTWARE FUNCTIONS

The voltmeter software is organized around a monitor routine awaiting input from the host via the IEEE-488 interface (as shown in Fig. 8). The input consists of ASCII strings of variable length--the first two characters being a code for setup or control. Strings are terminated by an end-or-identify (EOI) sent with the last character.

"Setup" codes are accompanied by an encoded byte string specifying the value for a certain parameter, such as the programmable gain for a specific A/D converter. Setup strings tell the program what values are to be used, but do not perform any operation such as physically setting the gain of an A/D converter. No default parameters are present within the program; all parameters must be specifically loaded.

"Control" codes (two-byte strings) command the voltmeter to perform certain functions; i.e., make an A/D conversion, compute certain parameters, transfer data to the host, etc.

The voltmeter can be requested to compute and return various parameters as requested, or to return the original sampled waveform to the host for special processing not supported by the voltmeter (plotting, for example). It is necessary to point out that although commands are transferred to the voltmeter as ASCII byte strings, data is received from the voltmeter in PDP-11 format (two bytes forming the 16-bit integers for sampled data, or four bytes floating point variables for all computed parameters). This was done in the interest of faster data transfer and maintaining original data precision for NRL-USRD applications. This feature prevents full compatibility with non-PDP 11 host processors, although subroutines to encode integer and floating-point variables exist as part of the display functions, and the program could be modified accordingly.

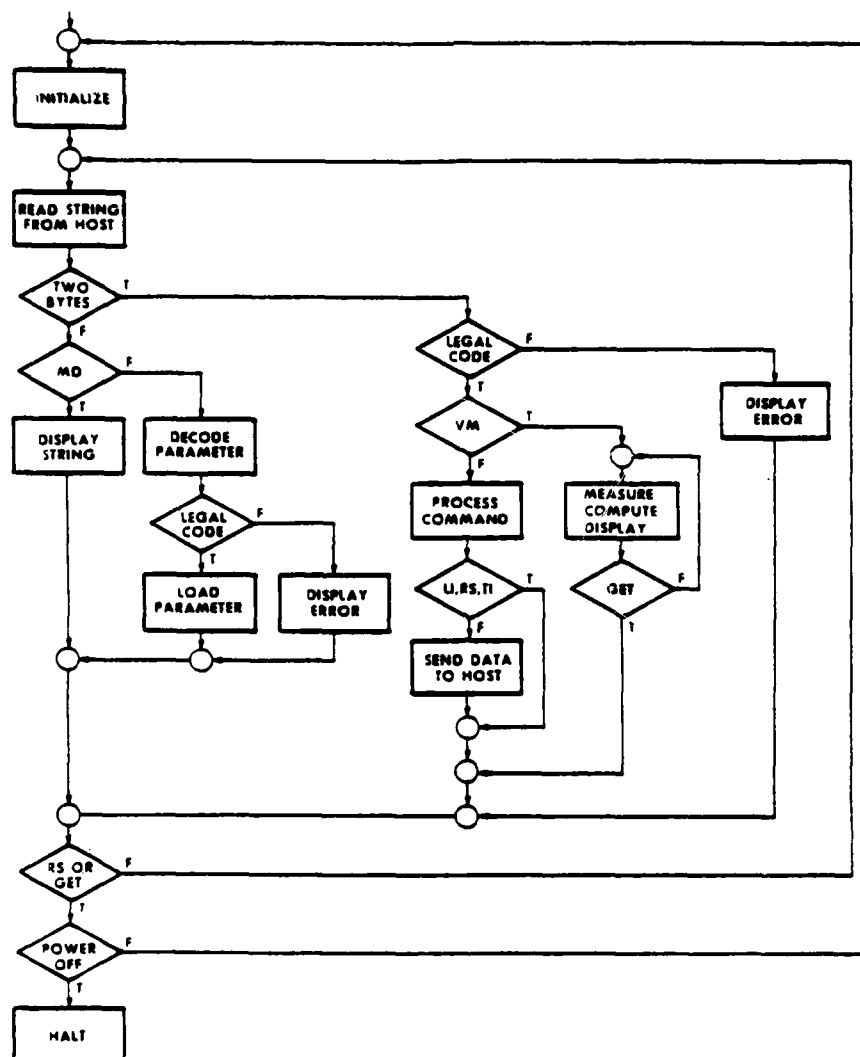


Fig. 9 - Flow chart of the basic monitor logic in the voltmeter software.

A special control code (VM) allows the voltmeter to measure, compute, and display as its own pace without host control; the host regains control by generating a group-execute-trigger (GET) on the IEEE-488 bus. The voltmeter displays data on either the console terminal or a 24-character plasma display. Display styles allowed are: count of A/D conversions and errors; voltage, phase, and distortion of one specific channel; peak voltage of all three channels; or voltages of two channels and the phase between the two.

The computation commands follow this sequence: the buffer of 12-bit digitized integer values is converted to floating-point numbers representing voltages within the A/D converter range (± 10 V); a window function is applied, if desired; the data sequence in the buffer is averaged into a smaller subsequence, if desired; the specific computation is made; the results are scaled to account for the A/D converter gain and corrected for amplitude

change due to the window function, if used; and the data are transmitted to the host. The use of windowing and subsequence averaging seems not very useful for the computation of nonsinusoidal parameters.

The four primary commands are: "D" (compute voltage, phase, and distortion from SFDFT); its subset, "C" (the same without distortion), used for more rapid computation where distortion is not needed; "E" (total rms, peak, and dc voltage); and a similar subset, "F" (absolute peak only), a rapid tool for system autoranging or overload detection. Relative computation times are shown in Table 3. Note that these times refer to a PDP-11/23 computer with both the KEF-11 floating-point processor and the new FPF-11 floating-point hardware. The FPF-11 offers as much as a factor of 5 improvement in computation speed.

Table 3 - Sample computations measured for a PDP-11/23 with the KEF-11 and FPF-11 floating-point processors.

COMPUTATION STYLE	COMPUTATION TIME IN SECONDS							
	NO SUBSEQUENCE AVERAGING				WITH SUBSEQUENCE AVERAGING			
	NO WINDOW		WITH WINDOW		NO WINDOW		WITH WINDOW	
	KEF-11	FPF-11	KEF-11	FPF-11	KEF-11	FPF-11	KEF-11	FPF-11
SINUSOIDAL PARAMETERS								
Voltage, Phase	0.494	0.149	0.570	0.172	0.210	0.098	0.293	0.132
Voltage, Phase, Distortion	2.627	0.506	2.632	0.531	0.383	0.133	0.458	0.165
NONSINUSOIDAL PARAMETERS								
rms, Peak, dc Voltage	0.350	0.151	-	-	0.198	0.100	-	-
Peak Voltage	0.189	0.096	-	-	-	-	-	-
Total Power	1.488	0.477	-	-	0.927	0.373	-	-

Average of 100 measurements of 20 cycles of data at 40 points per cycle (800 points). Distortion computed from first 7 harmonics. Subsequence averaging to 1 cycle. Times include data transfer on IEEE-488 bus (about 0.016 second).

Collateral computation commands include the ability to compute the total rms voltage and phase of a specific harmonic and to compute the power parameters second joint moment (ac and dc power) and covariance (ac power). The power parameters are the total power generated by all harmonics, and it is assumed that the host will scale the data properly to account for the calibration of the current measuring circuitry.

Further details of the software are to be found in the appendices and the liberally-commented source code located there. Appendix C lists various error and status messages; Appendix D gives listings for sample host routines to drive the voltmeter; and Appendix E includes complete listings of all software modules loaded into the voltmeter, with external discussion where appropriate.

SUMMARY

A multiple-channel microprocessor-based sampling voltmeter has been developed for use in the USRD measurement systems. This voltmeter is capable of measuring amplitude, phase, and distortion, and of using several computational algorithms to derive these results as the application requires. The original voltmeter has been used successfully in the Anechoic

Tank Facility as a high-voltage impedance measurement system, and this more powerful version has been greatly expanded to be used in Low-Frequency Facility, or other measurement systems as needed.

ACKNOWLEDGMENTS

The authors would like to acknowledge R.W. Anderson and R.W. Luckey, both of the USRD, and National Instruments, Inc., for their assistance in making this device function on the IEEE-488 interface bus, and C.K. Brown (USRD) for his assistance in incorporating the voltmeter into an operating electronic measurement system. This work was performed in the Acoustic Metrology Program sponsored by the Naval Sea Systems Command (SEA63R12).

REFERENCES

1. Barry A. Bell, Bruce F. Field, Thomas Kibalo, "A Fast Response Low-Frequency Sampling Voltmeter," National Bureau of Standards Tech Note 1159, National Bureau of Standards, Gaithersburg, MD, Aug 1982.
2. Dranetz Engineering Laboratories, Dranetz Model 3100 Sampling Network Analyzer Bulletin 3100, Edison, NJ.
3. G. Goertzel, "An Algorithm for the Evaluation of Finite Trig Series," *American Mathematical Monthly*, Vol. 65, 34-35 (1958).
4. G. Goertzel, *Mathematical Methods for Digital Computers, Vol I*, A. Ralston and H.S. Wolf, eds., John Wiley & Sons, New York, NY, 1960 (10th printing 1967), Ch. 24, 258-262.
5. F.J. Harris, "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform," *Proceedings of the IEEE*, Vol. 66, No. 1, 51-83 (1978).
6. J.F. Kaiser and R.W. Schafer, "On the Use of the I_0 -Sinh Window for Spectrum Analysis," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol ASSP-28, No. 1, 105-107 (1980).
7. Philip J. Davis and Philip Rabinowitz, *Numerical Integration*, Blaisdell Publishing Company, Waltham, Mass., 1967, Sec. 2.9.
8. National Instruments, "GPIB11V-1 Operating and Service Manual, April 1980, Part No. 320002-01; and "Software Reference Manual for National Instruments GPIB11-Series Interface Cards," September 1981, Part No. 310001-01.

APPENDIX A

Voltmeter Hardware

The hardware chosen as the heart of the voltmeter was the Digital Equipment Corporation (DEC) PDP-11/2 microcomputer (later upgraded to the PDP-11/23 with a KEF-11 floating-point processor), because it was a device that the USRD had both the experience and facilities to handle.

For A/D converters, ADAC Corp. models with DMA interfaces were chosen: model 1012, whose maximum sampling rate of 35 kHz was satisfactory for the LFF voltmeter, and model 1012AD, whose maximum sampling rate of 100 kHz was necessary for the ATF voltmeter.

Other interface modules used were:

MXV11-A	(DEC) Multifunction module: 2-serial line units 16K words RAM memory 4K words ROM memory (Intel 2732 EPROMs)
GPIBV11-1	(National Instruments) IEEE-488 bus interface
DRV11	(DEC) 16-bit parallel I/O card

The parallel I/O card is used only in pulsed systems to synchronize the measurements with the beginning of a toneburst.

(BLANK PAGE)

APPENDIX B

Voltmeter Commands

SETUP COMMANDS (# is channel number): These consist of a 2-byte code and a decimal integer string for the parameter value.

AE Number of times to automatically retry a measurement if an A/D error occurs

AI Which A/D-DMA device generates an interrupt when the data transfer is complete (1, 2, 3)

AN Which A/D converters are to be used; additive, where 1 = #1, 2 = #2, 4 = #3, (7 = all three)

AV Averaging mode (0 = no averaging, >0 = number of cycles in the averaged subsequence)

CT Computation type (for future use)

DD Display device: serial line 1 (console) or 2 (plasma display)

DS Display style: 0 (conversion count), 1-3 (voltage, phase, distortion of channels 1-3), 4 (peak voltage in all 3 channels), 5 (voltages in channels 1-2 and phase between)

FC Sample (clock) frequency in Hz

FS Signal frequency in Hz

GV Gated system (0 = not gated, 1 = use parallel I/O card to synchronize with toneburst)

G# Programmable gain for A/D converter (1, 2, 5, 10)

HA Number of harmonics used to compute distortion

HC Which harmonic to compute (see B# - CONTROL COMMANDS)

MD Transfer message to display, a byte string of up to 40 characters

NC Number of cycles per sequence

PA Number of points to convert per sequence

PR Number of points per sequence to return to host

PS First point to return to host. In practice, the first few converted points are bad, therefore, $PA = PR + PS$ points are converted and PR points are computed

P# Port (channel) for A/D Multiplexor (0-15)

WM Window mode (0 = don't use window, 1 = use window)

CONTROL COMMANDS (2-byte control codes; # is channel number)

B# Compute and return voltage and phase of the harmonic defined by HC parameter

CW Compute a sine wave for test purposes

C# Compute and return voltage and phase of the fundamental

D# Compute and return voltage, phase, and distortion of the fundamental

E# Compute and return statistical data (rms, dc, positive and negative voltage extremes)

F# Compute and return absolute peak voltage

LI List setup parameters on the display

ME Perform A/D conversion and return status

PO Compute and return total power (second joint moment)

RP Transfer setup parameters to host

RS Clear conversion and error counters

R# Read back a data sequence

RW Read back the average of a window and the weighted sequence

SJ Compute and return total power (second joint moment) and covariance (ac power)

TI Measure the time used in data transfer over the IEEE-488 bus

VM Set to free-running display voltmeter mode

WI Prepare voltmeter to read a computed window sequence from the host

SPECIAL IEEE-488 COMMANDS

- EOI End or identity - must coincide with last character sent by the host
- GET Group execute trigger - command will interrupt whatever the voltmeter is doing and reinitialize

APPENDIX C

Voltmeter Status Messages

Status is returned as an integer variable that is the voltmeter's acknowledgment of a measure command. It refers to either the legality of the command or the success of executing it.

<u>STATUS</u>	<u>MEANING</u>
1	Normal return
-1	A/D error in channel 1
-2	A/D error in channel 2
-4	A/D error in channel 3
-10	DMA error in channel 1
-20	DMA error in channel 2
-40	DMA error in channel 3
-100	A/D setup error
-101	Too many points for program buffer

Errors -1 through -40 are added together in case of multiple errors, and thus cover all numbers between -1 and -77.

The voltmeter processor will halt for the following severe errors:

<u>ADDRESS</u>	<u>REASON FOR HALT</u>
002036	Floating-point exception (can proceed)
002054	Function (sine, atan, sqrt) error (can proceed)
003370	Too many samples to compute (can proceed)
006224	Illegal display mode (can proceed)
013236	IEEE-488 interface illegally strapped (fatal)

(BLANK PAGE)

APPENDIX D

Voltmeter Control Subroutines

The following describes the protocol necessary and the available general-purpose subroutines for controlling this sampling voltmeter from a host processor. The National Instruments IEEE-488 bus driver and its software interface IBUP are a prerequisite [8].

Listings for the following subroutines (written at NRL-USRD) are included in this appendix.

1. Primitive subroutines

RDL SI	Poll the bus and if successful read a byte string from the voltmeter
WRL SI	Write a byte string to the voltmeter

2. Higher-level I/O subroutines

WRL PA	Send an integer parameter to voltmeter
WRL VM	Send an integer array to voltmeter
WR MEA	Send measurement command to voltmeter, wait an appropriate moment, enable specific clock synthesizer, receive status from voltmeter, disable specific clock synthesizer

3. Example of voltmeter I/O

MET ST	Test routine to command a measurement, read back a data sequence, and convert to volts
--------	--

```

0001      SUBROUTINE RDLSI (IADR,IBUF,ILEN,ISW)
          C
          C      SUBROUTINE TO READ A BYTE STRING FROM THE LSI VOLTMETER.
          C
          C      ARGUMENTS:
          C          IADR = GPIB BUS ADDRESS
          C          IBUF = BYTE-BUFFER TO LOAD WITH STRING
          C          ILEN = LENGTH OF STRING EXPECTED
          C          ISW = STATUS OF READ PROCESS
          C              ISW = 1 (SUCCESSFUL)
          C
          C      AUTHOR: R. E. SCOTT, JR. (NRL/USRD); VERSION: JANUARY 1982
          C
0002      BYTE IBUF(ILEN)
0003      INTEGER GPIB
0004      C
          C      100      CONTINUE
          C
          C      TEST FOR SRQ (CURRENTLY NOT WORKING WITH LSI)
          C      J=GPIB(13)
          C      IF (J.NE.1) WRITE (6,110) J
          C      CD110    FORMAT (' RDLSI TEST-SRQ FAILURE: '14)
          C      IF (J.NE.1) GOTO 100
          C
          C      POLL FOR RESPONSE
          C      J=IBUF (6,1ADR)
0005      IF (J.EQ.'102) GOTO 200 !SUCCESS
0006      IF (J.EQ.-6) GOTO 230 !TIMEOUT
0007      D
          D      WRITE (6,120) J
          D      D120    FORMAT (' RDLSI POLL FAILURE: '14)
0008      GOTO 100 !TRY AGAIN
          C
          C      READ THE BYTE STRING
          C      J=IBUF (1,IADR,IBUF,ILEN)
0009      200      IF (J.EQ.ILEN) ISW=1 !SUCCESS
0010      IF (J.NE.ILEN) ISW=J !FAILURE
0011      D
          D      WRITE (6,210) J,ISW
          D      D210    FORMAT (' RDLSI STATUS: J='13' ISW='13)
          D      WRITE (6,220) IBUF
          D      D220    FORMAT (' RDLSI STRING='1204)
          C
0012      RETURN
          C
0013      230      CONTINUE
          D      WRITE (6,240) J
          D      D240    FORMAT (' RDLSI TIMEOUT: J='13)
0014      RETURN
          C
0015      END

```

PROGRAM SECTIONS

Name	Size	Attributes
\$CODE1	000216	71 RW,1,CON,LCL
\$PDATA	000010	4 RW,D,CON,LCL
\$IDATA	000040	16 RW,D,CON,LCL
\$VARS	000004	2 RW,D,CON,LCL

Total Space Allocated = 000272 93

No FPP Instructions Generated

```

0001      SUBROUTINE WRLSI (IADR,STRING,ICNT,ISW)
          C
          C      SUBROUTINE TO WRITE A BYTE STRING TO THE LSI VOLTMETER.
          C
          C      ARGUMENTS:
          C          IADR -- BUS ADDRESS OF DEVICE
          C          STRING-- BYTE ARRAY TO BE SENT
          C          ICNT  -- LENGTH OF ARRAY TO BE SENT
          C          ISW   -- STATUS OF OPERATION
          C                      ISW=1 IS SUCCESSFUL
          C                      ISW=* UNSUCCESSFUL (* = GPIB ERROR)
          C
          C      AUTHOR: R. E. SCOTT, JR. (NRL/USRD); VERSION: JANUARY 1982
          C
0002      BYTE STRING(ICNT)
          C
          C      WRITE THE STRING
0003      J=IBUP(0,IADR,STRING,ICNT)
          C
          C      DID IT WORK? (IF ISW = ICNT, SUCCESS).
          C      IF NOT, ISW IS GPIB ERROR STATUS.
0004      IF (J.NE.ICNT) ISW=J
0005      IF (J.EQ.ICNT) ISW=1
          C
          C      WRITE (6,90) STRING
          C      D90  FORMAT (' WRLSI STRING='<ICNT>A1)
          C      D  WRITE (6,100) IADR,J,ISW
          C      D100 FORMAT (' WRLSI STATUS: ADDR='03' J='13' ISW='13')
          C
0006      RETURN
0007      END

```

PROGRAM SECTIONS

Name	Size	Attributes
\$CODE1	000134	46 RW,I,CON,LCL
\$PDATA	000004	2 RW,D,CON,LCL
\$IDATA	000032	13 RW,D,CON,LCL
\$VARS	000002	1 RW,D,CON,LCL

Total Space Allocated = 000174 62

No FPP Instructions Generated

```

0001      SUBROUTINE WRLPA (IADR,ICODE,IVALUE,ISW)
          C
          C      SUBROUTINE TO WRITE A CODE AND AN INTEGER VALUE TO
          C      THE LSI-VOLTMETER, AND READ AN ACKNOWLEDGEMENT.
          C      IADR = GPIB BUS ADDRESS OF VOLTMETER
          C      ICODE = TWO-BYTE PARAMETER SETUP CODE
          C      IVALUE = POSITIVE INTEGER PARAMETER TO LOAD
          C      ISW = STATUS OF OPERATION
          C      ISW = 1 (SUCCESSFUL AS ACKNOWLEDGED)
          C      ISW = * (FAILURE * AS ACKNOWLEDGED)
          C
          C      AUTHOR: R. E. SCOTT, JR. (NRL/USRD); VERSION: JANUARY 1982
          C
0002      BYTE STRING(8)
0003      BYTE ICODE(2)
          C
          C      INCORPORATE THE CODE INTO THE STRING
0004      STRING(1)=ICODE(1)
0005      STRING(2)=ICODE(2)
          C
          C      ENCODE PARAMETER INTO SIX-CHARACTER ASCII STRING
0006      ENCODE (6,100,STRING(3)) IVALUE
0007      100      FORMAT (I6)
          C
          C      REPLACE LEADING SPACES WITH ZEROS
0008      DO 110 I=3,8
0009      110      IF (STRING(I).EQ.' ') STRING(I)='0'
          C
0010      NB=8
          C
          C      WRITE THE STRING TO THE GPIB-BUS
          C      WRITE (6,200) IVALUE,STRING
          C      D200      FORMAT (' WRLPA VALUE='I6' STRING='8A1)
0011      CALL WRLSI (IADR,STRING,NB,ISW)
0012      IF (ISW.NE.1) CALL TSTERR (6,ISW)          !ERROR IS ???
          C      IF (ISW.EQ.-6) J=IBUP(10)          !TIMEOUT CLEANUP
          C      D      WRITE (6,300) ISW
          C      D300      FORMAT (' WRLPA (WRITE) STATUS: ISW='I3)
          C
          C      READ REPLY IF GIVEN
          C      CALL RDLSI (IADR,ISTAT,2,ISW)
          C      IF (ISW.NE.1) CALL TSTERR (6,ISW)          !ERROR IS ???
          C      IF (ISW.EQ.-6) J=IBUP(10)          !TIMEOUT CLEANUP
          C      CD      WRITE (6,310) ISW
          C      CD310      FORMAT (' WRLPA (READ) STATUS='I3)
          C
          C      SUBSTITUTE ACKNOWLEDGED STATUS FOR GPIB STATUS
          C      ISW=ISTAT
          C      RETURN
0013
0014      END

```

PROGRAM SECTIONS

Name	Size	Attributes
\$CODE1	000252	85
\$PDATA	000010	4
\$IDATA	000032	13
\$VARS	000014	6

Total Space Allocated = 000330 108

No FPP Instructions Generated

```

0001      SUBROUTINE WRLVM (IADR,ICODE,IVAL,ICNT,ISW)
          C
          C      SUBROUTINE TO WRITE A BYTE STRING TO THE LSI-VOLTMETER
          C      PREFIXED BY A TWO-BYTE PARAMETER CODE.
          C      IADR = GPIB BUS ADDRESS
          C      ICODE = TWO-BYTE PARAMETER CODE
          C      IVAL = ARRAY TO SEND TO VOLTMETER
          C      ICNT = NUMBER OF INTEGERS TO SEND
          C      ISW = STATUS OF ACKNOWLEDGEMENT
          C      IVAL COULD BE A 4-BYTE REAL VARIABLE (ICNT=2).
          C      IVAL COULD BE NOTHING (ICNT=0; SEND CODE ONLY).
          C
          C      AUTHOR: R. E. SCOTT, JR. (NRL/USDD); VERSION: JANUARY 1982
          C
0002      INTEGER ICODE
0003      INTEGER IVAL(8)
0004      INTEGER STRING(10)
          C
          C      PREFIX CODE INTO STRING
0005      STRING(1)=ICODE
0006      IF (ICNT.EQ.0) GOTO 60
          C
          C      ENTER VALUE INTO STRING, TWO BYTES AT A TIME
0007      DO 50 I=1,ICNT
0008      STRING(I+1)=IVAL(I)
          C
          C      COMPUTE STRING LENGTH
0009      60      NB=2+2*ICNT
          C
          C      WRITE THE STRING
0010      CALL WRLSI (IADR,STRING,NB,ISW)
0011      IF (ISW.NE.1) CALL TSTERR (6,ISW)
          C
          C      IF (ISW.EQ.-6) J=IBUP(10)
          C      IF (ISW.NE.1) WRITE (6,100) ISW
          C      100      FORMAT (' WRLVM STATUS: ISW='13)
          C
          C      READ REPLY IF GIVEN
          C      CALL RDLGI (IADR,ISTAT,2,ISW)
          C      IF (ISW.NE.1) CALL TSTERR (6,ISW)
          C      IF (ISW.EQ.-6) J=IBUP(10)
          C      IF (ISW.NE.1) WRITE (6,110) ISW
          C      110      FORMAT (' RDLGI STATUS: ISW='13)
          C
          C      EXCHANGE GPIB STATUS FOR ACKNOWLEDGEMENT STATUS
          C      ISW=ISTAT
          C      RETURN
0012
0013      END

```

!ERROR IS ???
!TIMEOUT CLEANUP

!ERROR IS ???
!TIMEOUT CLEANUP

PROGRAM SECTIONS

Name	Size	Attributes
\$CODE1	000234	78 RW,I,CON,LCL
\$PDATA	000034	2 RW,D,CON,LCL
\$IDATA	000032	13 RW,D,CON,LCL
\$VARS	000030	12 RW,D,CON,LCL
\$TEMP5	000002	1 RW,D,CON,LCL

Total Space Allocated = 000324 106

No FPP Instructions Generated

```

0001      SUBROUTINE WRMEA (IADR,ICODE,FREQ,ITIME,ISTAT)
          C
          C      SUBROUTINE TO WRITE A "ME" (MEASURE) TO THE LSI-VOLTMETER,
          C      (WAIT), ENABLE SAMPLING SYNTHESIZER, RECEIVE ACKNOWLEDGMENT,
          C      AND DISABLE SAMPLING SYNTHESIZER.
          C      *** SPECIFICALLY FOR LOW FREQUENCY SYSTEM SYNTHESIZER ***
          C
          C      IADR.....IEEE-488 BUS ADDRESS FOR VOLTMETER.
          C      ICODE.....MEASUREMENT CODE (ME).
          C      FREQ.....SAMPLING FREQUENCY IN HZ.
          C      ITIME.....TIME TO WAIT BETWEEN MEASURE COMMAND AND ENABLING
          C      SAMPLE-RATE GENERATOR (MSEC).
          C      ISTAT.....MEASUREMENT STATUS.
          C
          C      AUTHOR: R. E. SCOTT, JR. (NRL/USRD); VERSION: JANUARY 1982
          C
0002      ISTAT=0              !STATUS: NO MEASUREMENT
          C
          C      COMMAND A MEASUREMENT
0003      CALL WRLSI (IADR,'ME',2,ISW)
0004      IF (ISW.NE.1) CALL TSTERR (6,ISW) !BUS ERROR?
          D
          D100      IF (ISW.NE.1) WRITE (6,100) ISW
0005      FORMAT (' WRMEA WRITE-"ME" STATUS: '13)
          IF (ISW.NE.1) RETURN              !GIVE UP IF BUS ERROR
          C
0006      CALL WAIT (ITIME,1,K)              !WAIT FOR SLAVE TO BE READY
0007      CALL SM102 ('24,FREQ,'E ',JSW,ISW)!ENABLE SYNTHESIZER (CLOCK)
0008      CALL RDLIS (IADR,ISTAT,2,ISW)      !GET STATUS FROM SLAVE
0009      IF (ISW.NE.1) CALL TSTERR (6,ISW) !BUS ERROR?
          D
          D110      IF (ISW.NE.1) WRITE (6,110) ISW
0010      FORMAT (' WRMEA ACKNOWLEDGE STATUS: '13)
0011      CALL SM102 ('24,FREQ,'D ',JSW,ISW)!DISABLE SYNTHESIZER AGAIN
          RETURN
          C
0012      END

```

PROGRAM SECTIONS

Name	Size	Attributes
\$CODE1	000252	85
\$PDATA	000042	17
\$IDATA	000064	26
\$VARS	000006	3

Total Space Allocated = 000406 131

No FPP Instructions Generated

```

0001      SUBROUTINE METST (IAD,IT,MODE)

          C      SUBROUTINE TO TEST THE LEESBURG/RD-11/LSI-11/A22D
          C      HIGH-POWER MEASUREMENT SYSTEM INPUTS.

0002      INCLUDE 'LSPARM.COM'
0003      *      COMMON /LSPARM/SIFREQ,SIAMPL,SAFREQ,SAAMPL,
          *      IICAIN(3),NAD,NADT,NPAD,NPTR,NPST,NCYC,NHAR,
          *      IIAVGC,IIIDST,IICITYP,IIAERS,IIIGATE,IWINDO,IIDLUN,IHARC,
          *      IIPORT(3)
          *      C      SIFREQ,SIAMPL-----SIGNAL FREQUENCY AND AMPLITUDE
          *      C      SAFREQ,SAAMPL-----SAMPLE FREQUENCY AND AMPLITUDE
          *      C      IICAIN-----A/D PROGRAMMABLE GAIN
          *      C      NAD-----WHICH A/D'S USED (1=1,2=2,3=4,ALL=7)
          *      C      NADT-----WHICH A/D (DMA) TO INTERRUPT ON
          *      C      NPAD-----NUMBER OF POINTS TO CONVERT
          *      C      NPTR-----NUMBER OF POINTS TO TRANSFER
          *      C      NPST-----FIRST POINT TO TRANSFER
          *      C      NCYC-----NUMBER OF CYCLES TO CONVERT
          *      C      NHAR-----NUMBER OF HARMONICS USED TO COMPUTE DISTORTION
          *      C      IIAVGC-----NUMBER OF CYCLES TO AVERAGE TO (0=DON'T)
          *      C      IIIDST-----DISPLAY STYLE (0--5)
          *      C      IICITYP-----COMPUTATION TYPE
          *      C      IIAERS-----A/D ERRORS TO RETRY BEFORE QUITTING
          *      C      IIIGATE-----GATED (1) OR NOT (0) ((DRV-11 IN USE))
          *      C      IWINDO-----MULTIPLY DATA BY WINDOW (WDATA)
          *      C      IIDLUN-----DISPLAY DEVICE
          *      C      IHARC-----HARMONIC TO COMPUTE

0004      INCLUDE 'LSDATA.COM'
0005      *      PARAMETER BSIZE=1024
          *      C      PARAMETER NOAD=3
          *      C      COMMON /IDAT/IDATA(NOAD*BSIZE)
          *      C      COMMON /RDAT/RDATA(NOAD*BSIZE)
          *      C      COMMON /WDAT/WDATA(BSIZE),WATT,WAV
          *      C      COMMON /SDAT/SDATA(BSIZE)
          *      C      COMMON /RTIM/TIM(NOAD)
          *      C      !MAXIMUM POINTS TO READ
          *      C      !NUMBER OF CHANNELS
          *      C      !RAW A/D DATA (IN BITS)
          *      C      !NORMALIZED DATA (IN VOLTS)
          *      C      !WINDOW, ATTENUATION, AVERAGE
          *      C      !COMPUTATION BUFFER
          *      C      !COMPUTATION TIMES

0012      INTEGER IIPAR(15),IIAE(6)

0013      IF (IT.EQ.'ME') GOTO 100
0014      IF (IT.EQ.'RI') GOTO 200
0015      IF (IT.EQ.'R2') GOTO 200
0016      IF (IT.EQ.'R3') GOTO 200

          C
          C      GO TO LS1 TO PERFORM A/D CONVERSION (ME)
0017      100 CONTINUE
          C      WRITE (6,101)
0018      2101 FORMAT (' DO A/D CONVERSION')
          C      IF (MODE.NE.'LO') CALL WRLVM (IAD,'ME',0.0,ISW)
          C      IF (MODE.EQ.'LO') CALL WRMEA (IAD,'ME',SAFREQ,200,ISW)
          C      IF (ISW.GE.0) RETURN
          C      WRITE (6,102) ISW
0022      102 FORMAT (' *** A/D CONVERSION ERROR: '13' ***')
          C      IF (ISW.EQ.-100) WRITE (6,110)
0023      IF (ISW.EQ.-101) WRITE (6,120)
          C      IF (ISW.EQ.-101) WRITE (6,120)
0025      110 FORMAT (' A/D SETUP ERROR (A/D'S OR INTERRUPT)')
          C      IF (ISW.EQ.-100) WRITE (6,120)
0026      120 FORMAT (' TOO MANY POINTS REQUESTED')
          C      RETURN
0027      C
          C      READ A DATA BUFFER FROM THE LSI (R1,R2,R3)
0028      200 CONTINUE
          C      WRITE (6,201)
          C      0201 FORMAT (' READ FROM 11/03')
          C      RDS=SECONDS(0.0)
0029      CALL WRLVM (IAD,IT,0.0,ISW)
0030

```

```

0031      IF (ISW.NE.1) WRITE (6,203) ISW
0032      203      FORMAT (' WRLVM ERROR: '13)
0033      IF (ISW.EQ.-6) J=IBUP(10)
0034      IF (ISW.NE.1) RETURN
0035      IF (IT.EQ.'R1') J=1
0036      IF (IT.EQ.'R2') J=2
0037      IF (IT.EQ.'R3') J=3
0038      K=(J-1)*NPTR+1
0039      CALL RDLS1 (IAD,1DATA(K),2*NPTR,ISW)
      C
      220      WRITE (6,220) (1DATA(I),I=K,K+NPTR)
      220      FORMAT (' '808)
0040      IF (ISW.NE.1) WRITE (6,230) J,ISW
0041      230      FORMAT (' *** A/D DATA CHANNEL '11' IN ERROR: '13' ***')
0042      IF (ISW.EQ.-6) J=IBUP(10)
0043      IF (ISW.NE.1) RETURN
      C
      C      FIX THE DATA PROPERLY
      LI=1+(J-1)*NPTR
      L2=L1+NPTR
      DO 250 I=L1,L2
      IF=IAND('7777',1DATA(I))
      IF (IF.GE.'4000') IM=IOR('170000,IM)
      1DATA(I)=10.0*IM/'4000
0049      250      CONTINUE
0050      TIM(J)=SECNDS(RDS)
0051      D      IF ((IT.EQ.'R1').OR.(IT.EQ.'R2')) WRITE (6,260) TIM(J)
      D      IF ((IT.EQ.'R3').OR.(IT.EQ.'RE')) WRITE (6,260) TIM(J)
      260      FORMAT (' BUFFER READ TIME='F8.3)
      C
0052      RETURN
0053      END

```

PROGRAM SECTIONS

Name	Size	Attributes	
\$CODE1	001156	311	RW,I,CON,LCL
\$PDATA	000324	106	RW,D,CON,LCL
\$1DATA	000056	23	RW,D,CON,LCL
\$VARS	000074	30	RW,D,CON,LCL
LSPARM	000072	29	RW,D,OVR,GBL
1DAT	014000	3072	RW,D,OVR,GBL
RDAT	030000	6144	RW,D,OVR,GBL
WDAT	010010	2052	RW,D,OVR,GBL
SDAT	010000	2048	RW,D,OVR,GBL
1TIM	000014	6	RW,D,OVR,GBL

Total Space Allocated = 065772 13821

APPENDIX E

Voltmeter Program Listings

The modules used in this program are written in PDP-11 assembly language, divided into pure (ROM) code (modules 1-7) with all dynamic variables (RAM) in the final module.

- | | | |
|-----|--------|--|
| (1) | LSIPL | Command interpreter; measurement, computation, and display driver |
| (2) | AD2D | A/D converter control subroutine |
| (3) | CONVRT | Floating to ASCII encoding subroutine, used to prepare numbers for display or transfer to the host |
| (4) | DAPD | Computation of amplitude, phase, distortion; uses DGZL below to obtain real and imaginary parts of voltage |
| (5) | DGZL | Single frequency DFT subroutine: computes real and imaginary parts of voltage of a given spectral line by Goertzel's algorithm |
| (6) | GPIOSR | IEEE-488 bus input and output subroutines, initialize subroutine, and interrupt service routine |
| (7) | GRMS | Computation by time-integration of rms, dc, positive peak, voltages |
| (8) | LSIVAR | All the dynamic variables used by (1)-(6) above |

Also used are modules from DEC's RSX-11M System Library--\$ATAN (arctangent), \$SIN (sine), and \$SQRT (square root).

The subroutines (2)-(7) are written to conform with DEC standards; argument lists are passed through register 5 so that these routines could be used by other programs. Conditional assembly parameters allow dynamic variables to be assembled locally, or in an external global file such as LSIVAR.

- (1) MODULE LSIVM - Main program for controlling the microcomputer-controlled sampling voltmeter

Internal Summary:

<u>(LABEL)</u>	<u>(FUNCTION)</u>	<u>(PAGE)</u>
<u>MONITOR ROUTINES</u>		
START	Initialize system	5
RDCODE	Read string from bus; interpret type of command	5
<u>PARAMETER ROUTINES</u>		
PARAM	Parameter decode and load	6
FREQP	Read frequency data	8
MESDAY	Read and display text string	8
DECOD	Byte string to integer decoding	9
FPISR	Floating-point processor interrupt service routine	9
EXEC	Interpret and direct control codes	10
CERR	Parameter I/O error routine	11
<u>CONTROL ROUTINES</u>		
ABORT	Abort process and reinitialize	11
MESURE	Make measurement	12
WAKE	Counter/display initialization	12
PARLST	Display setup parameters	13
IFWAIT	Delay subroutine	13
PARSND	Send setup parameters to host	14
ADSEND	Send A/D buffer or window to host	14
<u>PRE-PROCESSING ROUTINES</u>		
CMPUT	Computation interpreter	15
CWAVE	Compute test sine wave	17
FLOT	Convert A/D buffer to floating point buffer	18

<u>(LABEL)</u>	<u>(FUNCTION)</u>	<u>(PAGE)</u>
WINDIN	Read sequence window from host; compute average	19
WINDOW	Multiply data buffer by window buffer	19
BAVER	Average data buffer to fewer cycles	20
<u>COMPUTATION ROUTINES</u>		
GDFTD	Compute voltage, phase, distortion (DFT) of fundamental	21
GDFT	Compute voltage, phase (DFT) of fundamental	21
GSL	Compute voltage, phase (DFT) of harmonic	21
TDFTD	Transfer voltage, phase, distortion to host	22
TDFT	Transfer voltage, phase to host; display	22
DSLD	Load DFT values for display	22
GRMSV	Do statistical computation	23
GPEKO	Find absolute peak value	23
TRMSV	Transfer statistical data to host; display	24
TPEK	Transfer peak voltage to host; display	24
DEGAIN	Correct voltage for A/D gain	24
DEWIND	Correct DFT voltage for window average	24
GPWR	Compute total power (second joint moment)	25
GJC	Compute total power and ac power	26
GMN	Get rms voltage for power computation	26
TJC	Transfer total power and ac power to host	26
<u>DISPLAY ROUTINES</u>		
VMDISP	Stand-alone voltmeter executive	27
CHODIS	Display, mode 0 (conversion and error count)	29
CHIDIS	Display, modes 1-3 (voltage, phase, and distortion)	30

CH4DIS	Display, mode 4 (voltages)	32
CH5DIS	Display, mode 5 (voltages and relative phase)	33
WRITE	Terminal output (string)	34
PRINT	Terminal output (integer)	34

```

1  TITLE LSIVH
2  IDENT /06/
3
4
5
6  ;NRL/USRD MICROCOMPUTER SAMPLING VOLTMETER PROGRAM.
7  ;AUTHOR: RICHARD E. SCOTT, JR. (FEBRUARY 1982)
8
9
10
11
12  ;THIS PROGRAM IS DESIGNED TO PERFORM AS A SAMPLING VOLTMETER
13  ;CONTROLLED BY THE GPIB INTERFACE BUS. THE SETUP AND CONTROL
14  ;COMMANDS ARE SENT IN MULTI-BYTE PACKETS. DATA IS RETURNED
15  ;ON COMMAND IN THE FORM OF A WAVEFORM DIGITIZED BY UP TO THREE
16  ;A/D CONVERTERS, AND AS VARIOUS COMPUTED PARAMETERS.
17
18  ;THE "MEASURE" COMMAND IS EFFECTIVELY THE ONLY ONE THAT IS
19  ;ACKNOWLEDGED BY THIS PROGRAM, RETURNING AN INTEGER STATUS.
20
21  ;THE CURRENT VERSION CAN RETURN THE FOLLOWING PARAMETERS
22  ;UNDER HOST CONTROL: RMS, PEAK, AND DC VOLTAGE FROM TIME
23  ;INTEGRATION; RMS VOLTAGE, PHASE, AND DISTORTION FROM A FFT
24  ;CALCULATION. ALSO RETURNED IS TOTAL POWER (POWER IN ALL
25  ;HARMONICS). PREPROCESSING OPTIONS INCLUDE MULTIPLYING THE
26  ;DATA SEQUENCE BY A WINDOW SPECIFIED BY THE HOST, AND
27  ;AVERAGING MANY-CYCLED SEQUENCES DOWN TO FEW CYCLES WHERE
28  ;SUCH IS PERMISSABLE.
29
30  ;THE PROGRAM CONTAINS SEPARATE, CONTIGUOUS DATA BUFFERS OF A
31  ;GIVEN MAXIMUM SIZE FOR EACH OF THE THREE CHANNELS. HOWEVER,
32  ;DATA IS DIGITIZED INTO MEMORY AS IF THESE ARRAYS WERE CONTIG-
33  ;UOUS, BUT SIZED FOR THE NUMBER OF SAMPLES DIGITIZED. THE
34  ;COMPUTATION IS PERFORMED ON ONE CHANNEL'S DATA AT A TIME, AS
35  ;EACH INTEGER SEQUENCE MUST FIRST BE CONVERTED TO FLOATING-POINT
36  ;VALUES STORED IN THE SINGLE "TEMPORARY" FLOATING ARRAY. TOTAL
37  ;POWER, REQUIRING TWO FLOATING-POINT CHANNELS SIMULTANEOUSLY,
38  ;IS COMPUTED DIFFERENTLY (A SAMPLE AT A TIME).
39
40  ;ALTHOUGH THIS DEVICE CAN ACT LIKE A DISPLAY VOLTMETER
41  ;INDEPENDENT OF COMPUTER CONTROL, IT IS STILL NECESSARY
42  ;TO SET UP THE DEVICE UNDER COMPUTER/BUS CONTROL, BECAUSE
43  ;IT IS FREQUENCY-DEPENDENT, NO DEFAULT VALUES ARE PRACTICAL
44  ;TO ALLOW IT TO POWER-ON INTO A VOLTAGE-READING MODE.
45
46  ;THE VOLTMETER IS CONTROLLED BY COMMANDS FROM THE BUS,
47  ;WHICH CONSIST OF A TWO-BYTE (TWO CHARACTER) CODE, AND
48  ;AN INTEGER ARGUMENT (IF NEEDED) AS A BYTE STRING.
49
50  ;ANY PROCESS CAN BE PERMANENTLY ABORTED FROM THE HOST
51  ;BY TRANSMITTING A DEVICE TRIGGER (GET) ON THE BUS.
52  ;ALL DEVICES AND COUNTERS ARE RESET.

```

```

53 ;SETUP COMMANDS (TWO BYTES PLUS BYTE STRING)-
54 ;FORMAT (TWO-BYTE CODE) (MULTI-BYTE VALUE)
55 ; C: PROGRAMMABLE GAIN FOR A/D CONVERTER # (1-3) (1,2,5,10)
56 ; P: INPUT PORT FOR A/D CONVERTER # (1-3) (0-16)
57 ; AN: WHICH OF A/D'S USED (#1-1, #2-2, #3-4; ALL=7)
58 ; AI: WHICH A/D TO INTERRUPT ON (DMA FINISH)
59 ; AE: HOW MANY A/D ERRORS TO RETRY BEFORE GIVEUP
60 ; PA: NUMBER OF POINTS (CONVERSIONS) PER WAVEFORM SEQUENCE
61 ; PR: NUMBER OF POINTS (CONVERSIONS) TO RETURN TO HOST
62 ; PS: FIRST POINT (CONVERSION) NUMBER TO RETURN TO HOST
63 ; NG: NUMBER OF CYCLES PER SEQUENCE
64 ; HA: NUMBER OF HARMONICS TO USE COMPUTING DISTORTION
65 ; HC: WHICH HARMONIC TO COMPUTE
66 ; DS: DISPLAY TYPE (Q.V.)
67 ; DD: DISPLAY DEVICE (SEE PAGE 3)
68 ; AV: AVERAGING MODE (0=NONE, N=CYCLES RESULTING)
69 ; CT: COMPUTATION TYPE (Q.V.)
70 ; CV: GATED (0=NO, 1=YES)
71 ; WN: WINDOW MODE (0=NONE, 1=USE)
72 ; MD: SEND MESSAGE TO DISPLAY
73
74
75 ; INFORMATIONAL COMMANDS (SIX BYTES)-(OBSOLETE)
76 ;FORMAT (TWO-BYTE CODE) (FOUR-BYTE REAL VALUE)
77 ; FS: SIGNAL FREQUENCY IN HZ
78 ; FC: CLOCK FREQUENCY IN HZ
79
80
81 ; CONTROL COMMANDS (TWO BYTES)-
82 ;FORMAT (TWO-BYTE CODE)
83 ; RS: CLEAR COUNTERS
84 ; NE: PERFORM A/D CONVERSION AS PROGRAMMED
85 ; LI: LIST 17 PARAMETERS ON THE LSIVM CONSOLE
86 ; VN: GO TO VOLTMETER MODE (AUTOMATIC)
87 ; WI: INPUT WINDOW DATA ("PR" REAL VALUES, 4*PR BYTES)
88 ; TI: DUMMY COMMAND FOR I/O TIMING INFORMATION
89 ; CW: COMPUTE SINE WAVEFORM FOR TIMING INFORMATION
90
91
92 ; DATA TRANSFER COMMANDS:
93 ;FORMAT (TWO-BYTE CODE)
94 ; H: TRANSFER WAVEFORM # (1-3) SAMPLES BACK TO HOST
95 ; RW: TRANSFER WINDOW AND AVERAGE BACK TO HOST
96 ; RP: TRANSFER 16 PARAMETERS BACK TO HOST FOR CHECKING
97 ; ALSO.....
98
99 ; COMPUTE COMMANDS (TWO BYTES)-
100 ;FORMAT (TWO-BYTE CODE)
101 ; B: RETURN VOLTAGE AND PHASE OF HCTH HARMONIC, CHANNEL # (1-3)
102 ; G: RETURN VOLTAGE AND PHASE OF FUNDAMENTAL, CHANNEL # (1-3)
103 ; D: RETURN VOLTAGE, PHASE, DISTORTION (FROM DFT), CHANNEL # (1-3)
104 ; E: RETURN RMS, PEAK+, PEAK-, DC VOLTAGES, CHANNEL # (1-3)
105 ; F: RETURN PEAK ONLY, CHANNEL # (1-3)
106 ; P0: RETURN TOTAL POWER FROM CHANNEL #1 (VOLTS) AND #2 (AMPS)
107 ; CJ: RETURN SECOND JOINT MOMENT AND COVARIANCE, CHANNELS #1 AND #2
108

```

```

110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158

SUBROUTINES REQUIRED:
:
: GPIOSR -- GPIP-488 BUS CONTROL SUBROUTINES
: A32D -- ADAC A/D CONVERTER CONTROL SUBROUTINE
: CONVERT -- READ TO ASCII ENCODING FOR DISPLAY
: GUNS -- COMPUTE TANK-SERIES VOLTAGES
: DGZL -- COMPUTE DISTANCE, PHASE AND DISTANCE
: GAPD -- COMPUTE DISTANCE, PHASE AND DISTANCE
: LSIVAR -- GLOBAL DATA VARIABLES
:
SPECIAL ASSEMBLY INSTRUCTIONS:
:
: THIS SYSTEM WAS DESIGNED TO BE LOADED INTO PROM, SO THE
: BAR VARIABLES ARE CONCENTRATED IN THE MODULE LSIVAR. IT
: IS NECESSARY TO CHECK THE STATUS OF THE CONDITIONAL
: ASSEMBLY PARAMETER "PROM" IN THE MAIN ROUTINE LSIVM AND
: MANY OF THE SUBROUTINES TO INSURE THAT IT IS IN THE RIGHT
: STATE. IN LSIVM PROM CONCERNS WHETHER PAGE ZERO (VECTOR
: SPACE) IS IN PROM (HOLD ONLY NO); IN A32D PROM CONCERNS
: WHETHER VECTORS ARE TO BE LOGICALLY LOADED; IN DGZL AND
: GAPD IT CONCERNS WHETHER BAR VARIABLES ARE LOGICALLY ASSEMBLED
: (AS FOR OTHER USERS), OR IN LSIVAR (THIS APPLICATION).
:
SPECIAL TASK-BUILD INSTRUCTIONS:
:
: DESIGNED ONLY TO RUN ON A MICRO-ALONE PDP-11, AND WILL NOT
: EXECUTE UNDER MCA-11C. IT IS NOT RELOCATABLE.
:
: FOR COMMAND FILE:
: LSIVM/SQ/-HM/-HD/FP,LSIVM,GP SP=LSIVM
: GUNS,A32D,CONVRT,DGZL,GAPD
: GPIOSR
: 11.11SYSLIB,OBJ,BUILD,CATAM:$SIN:$DSIN:$DSOUT
: LSIVAR
:
: STACK=0
: UNITS=0
: PAR-PAR:1000:1000
: ACTFL=0
: A32D=0
: DTPL=0
:
AC0=30
AC1=21
AC2=22

000000
000001
000002

```

```

;TEXT PRINTING ADDRESSES FOR SERIAL LINE INTERFACES
DIAXSR=177564 ;CONSOLE (DLUN = 1)
DIAXBF=177566
CHXSX=176504 ;DISPLAY (DLUN = 2)
CHXBF=176506

;CPIB REGISTERS REQUIRED
CPACR=160096 ;CPIB AUXILIARY COMMAND REGISTER

;SET PROM = 1 IF VECTORS TO BE LOADED IN PROM.
;IF SO ALSO CHANGE TASKBUILD FILE TO PAR=PAR:0 ....
PROM = 1

```

LINE	ADDRESS	DATA	COMMENT
160	177564		
161	177566		
162	177566		
163	176504		
164	176506		
165			
166	160096		
167			
168			
169			
170	000001		
171			
172			
173			
174			
175	000167	000774	
176	000000		
177	176	000004	
178	000024	001000	
179	000026	000340	
180	000030		
181	000034	002054	
182	000036	000340	
183	000040		
184	000040		
185	000040		
186	000040		
187	000040		
188	000040		
189	000040		
190	000040		
191	000040		
192	000040		
193	000040		
194	000040		
195	000040		
196	000040		
197	000040		
198	000040		
199	000040		
200	000040		
201	000040		
202	000040		
203	000040		
204	000040		
205	000040		
206	000040		
207	000040		
208	000040		
209	000040		
210	000040		
211	000040		
212	000040		
213			

.ENDC

```

215 ;START OF PROGRAM
216 ;INITIALIZE WHATEVER IS NECESSARY AND GO INTO "WAIT" STATE.
217 ;NORMAL STATUS IS TO WAIT FOR INPUT FROM THE IEEE-488 BUS;
218 ;WHEN RECEIVED, GO TO EITHER DATA OR COMMAND INTERPRETER.
219
220
221 001006 012766 00000000
222 001000 012766 00000005
223 001004 0000005
224
225
226
227
228
229
230 001006 005037 00000000
231 001012 005037 00000000
232 001016 012763 001112
233 001022 004767 00000000
234 001026 012700 00000000
235 001032 106400
236
237
238
239
240
241 001034
242 001034 312767 000001 00000000
243 001042 012705 001102
244 001016 004767 00000000
245
246 001052 023727 00000000 000002
247 001060 001002
248 001062 000167 000772
249
250 001066 022737 042115 00000000 100
251 001074 001012
252 001076 000167 000626
253
254 001102 000093 00000000 001116 GPIARG: 3, INBUF, MLEN, IBLEN
255 001110 00000000
256 001112 000001 002410
257 001116 000050
258 001120 00000000

```

START:

```

MOV $STACK, SP
RESET
IF EQ PROM
MOV $FPISR, 0#244
MOV $340, 0#246
MOV $F0ISR, 0#34
MOV $340, 0#36
ENDC
CLR 0#1CA
CLR 0#1CE
MOV $CPINA, R5
JSR PC, GPIN1
MOV $0, R0
HTTS R0

```

INITIALIZE STACK POINTER
CLEAR Q-BUS
LOAD FLOATING-ERROR INTERRUPT VECTOR
AND PSW
LOAD FUNCTION-ERROR INTERRUPT VECTOR
AND PSW
NUMBER OF A/D CONVERSIONS
NUMBER OF A/D ERRORS
INITIALIZE IEEE-488 BUS CARD
CLEAR PSW
ENABLE INTERRUPTS

THIS PORTION OF THE PROGRAM AWAITS A DATA PACKET FROM THE
HOST PROCESSOR AND DIRECTS IT TO THE PROPER FUNCTION.

DEFAULT STATUS = 1
READ A CODE FROM THE GPIB
HOW MANY BYTES?
IF TWO, EXECUTE A COMMAND
OTHERWISE INTERPRET DATA
CODE = "RD"?
NO
YES, REQUIRES SPECIAL PROCESSING

MAXIMUM INPUT STRING LENGTH
MAXIMUM DATA BUFFER SIZE

RD CODE:

```

MOV $1, ISW
MOV $GPIARG, R5
JSR PC, GPIN
CMP $IBLEN, $2
BNE 100
EXEC
CMP $RD, 0#1C
BNE PARAM
JMP MESDAY

```

1.ABORT
WORD 40
WORD RSIZE

```

259      ;THIS PORTION OF THE PROGRAM HANDLES THE INPUT OF DATA
260      ;PARAMETERS. RECEIVED BYTE STRING IS DECODED INTO AN
261      ;INTEGER VARIABLE IN R3, AND THEN LOADED INTO THE VARIABLE
262      ;SPECIFIED BY THE CODE WORD IC.
263
264      001122      004767      000634      PARAM:
265      001122      022737      030507      0000000C      PC, DECOD
266
267      001126      022737      030507      0000000C      ;CODE = "G1"?
268      001134      001004
269      001136      010337      0000000C      ;GAIN FOR CHANNEL 1
270      001142      000167      177666      RDCODE
271
272      001146      022737      031107      0000000C 10$ :
273      001154      001004
274      001156      010337      0000000C      ;CODE = "G2"?
275      001162      000167      177646      ;GAIN FOR CHANNEL 2
276
277      001166      022737      031507      0000000C 20$ :
278      001174      001004
279      001176      010337      0000000C      ;CODE = "G3"?
280      001202      000167      177626      ;GAIN FOR CHANNEL 3
281
282      001206      022737      047101      0000000C 30$ :
283      001214      001004
284      001216      010337      0000000C      ;CODE = "AN"?
285      001222      000167      177606      ;NUMBER OF A/D'S IN USE
286
287      001226      022737      044501      0000000C 40$ :
288      001234      001004
289      001236      010337      0000000C      ;CODE = "A1"?
290      001242      000167      177566      ;WHICH A/D TO INTERRUPT ON (DMA FINISH)
291
292      001246      022737      040520      0000000C 50$ :
293      001254      001004
294      001256      010337      0000000C      ;CODE = "PA"?
295      001262      000167      177546      ;SAMPLES TO CONVERT PER DMA
296
297      001266      022737      051120      0000000C 60$ :
298      001274      001004
299      001276      010337      0000000C      ;CODE = "PR"?
300      001302      000167      177526      ;NUMBER SAMPLES TO RETURN TO HOST
301
302      001306      022737      051520      0000000C 70$ :
303      001314      001004
304      001316      010337      0000000C      ;CODE = "PS"?
305      001322      000167      177506      ;FIRST SAMPLE TO RETURN TO HOST
306
307      001326      022737      041516      0000000C 80$ :
308      001334      001004
309      001336      010337      0000000C      ;CODE = "NC"?
310      001342      000167      177466      ;NUMBER CYCLES PER SEQUENCE
311
312      001346      022737      040510      0000000C 90$ :
313      001354      001004
314      001356      010337      0000000C      ;CODE = "HA"?
315      001362      000400      ;HOW MANY HARMONICS TO PROCESS

```

317	001364	022737	053161	0000000 2000	CMP	*"AV, @*IC	ICODE = "AV"?
318	001372	001004			BNE	200	
319	001374	010337	0500000		MOV	R3, @*AVCCOD	HAVERAGING PROCESS TO FOLLOW
320	001400	000167	177410		JMP	RDCODE	
321							
322	001404	022737	051504	0000000 2000	CMP	*"DS, @*IC	ICODE = "DS"?
323	001412	001004			BNE	300	
324	001414	010337	0000000		MOV	R3, @*DSTYLE	DISPLAY STYLE
325	001420	000167	177410		JMP	RDCODE	
326							
327	001424	022737	051107	0000000 3000	CMP	*"CV, @*IC	ICODE = "CV"?
328	001432	001004			BNE	400	
329	001434	010337	0000000		MOV	R3, @*GATED	GATING FLAG
330	001440	000167	177370		JMP	RDCODE	
331							
332	001444	022737	042501	0000000 4000	CMP	*"AE, @*IC	ICODE = "AE"?
333	001452	001004			BNE	500	
334	001454	010337	0000000		MOV	R3, @*AERES	RETRY ERRORS COUNTER
335	001460	000167	177350		JMP	RDCODE	
336							
337	001464	022737	052103	0000000 5000	CMP	*"CT, @*IC	ICODE = "CT"?
338	001472	001004			BNE	600	
339	001474	010337	0000000		MOV	R3, @*CORTYP	COMPUTATION TYPE
340	001500	000167	177330		JMP	RDCODE	
341							
342	001504	022737	042104	0000000 6000	CMP	*"DD, @*IC	ICODE = "DD"?
343	001512	001004			BNE	700	
344	001514	010337	0000000		MOV	R3, @*DLUN	DISPLAY LUN
345	001520	000167	177310		JMP	RDCODE	
346							
347	001524	022737	046527	0000000 7000	CMP	*"WH, @*IC	ICODE = "WH"?
348	001532	001004			BNE	800	
349	001534	010337	0000000		MOV	R3, @*MODE	WINDOW MODE
350	001540	000167	177270		JMP	RDCODE	
351							
352	001544	022737	041510	0000000 8000	CMP	*"RC, @*IC	ICODE = "RC"?
353	001552	001004			BNE	900	
354	001554	010337	0000000		MOV	R3, @*HTC	HARMONIC TO COMPUTE
355	001560	000167	177250		JMP	RDCODE	
356							
357	001564	022737	030520	0000000 9000	CMP	*"P1, @*IC	ICODE = "P1"?
358	001572	001004			BNE	1000	
359	001574	010337	0000000		MOV	R3, @*AD1C	A/D CHANNEL 1 PORT
360	001600	000167	177230		JMP	RDCODE	
361							
362	001604	022737	031120	0000000 10000	CMP	*"P2, @*IC	ICODE = "P2"?
363	001612	001004			BNE	1100	
364	001614	010337	0000000		MOV	R3, @*AD2C	A/D CHANNEL 2 PORT
365	001620	000167	177210		JMP	RDCODE	
366							
367	001624	022737	031520	0000000 11000	CMP	*"P3, @*IC	ICODE = "P3"?
368	001632	001004			BNE	1200	
369	001634	010337	0000000		MOV	R3, @*AD3C	A/D CHANNEL 3 PORT
370	001640	000167	177170		JMP	RDCODE	
371							

(THIS PORTION WILL READ THE "REAL" PARAMETERS, SUCH AS
 SIGNAL AND SAMPLE FREQUENCY; THIS IS NO LONGER USED.
 DATA INPUT IS FOUR BYTES WHICH ARE MORE LITERALLY A
 32-BIT FLOATING VARIABLE IN PDP-11 FORMAT.

```

373
374
375
376
377
378 001644      FREMP:
379 001644      022737 051506 000000C
380 001652      001007
381 001654      013737 000000C 000000C
382 001662      013737 000002C 000002C
383 001670      000115
384
385 001672      022737 041506 000000C 10$
386 001709      001907
387 001702      013737 000000C 000000C
388 001710      013737 000002C 000002C
389 001716      000402
390
391
392 001720      000167 20$
393
394 001724      000167 177104 100$
395
396
397
398
399
400
401 001730      MESDAY:
402 001730      012705 001752
403 001734      012737 006412 000000C
404 001742      004767 006222
405 001746      000167 177062
406
407 001752      000002 000000C MESARG: 2,1C,1BLEN,BLEN
408 001760      000000C

```

```

CMP      *FS,*FC      ;IS THE CODE "FS"?
BNE      10$
MOV      *IP,*FSIG     ;SIGNAL FREQUENCY
MOV      *IP+2,*FSIG+2
BR       100$

```

```

CMP      *FC,*FC      ;IS THE CODE "FC"?
BNE      20$
MOV      *IP,*FSAM     ;SAMPLE FREQUENCY
MOV      *IP+2,*FSAM+2
BR       100$

```

```

IMP      CEHR          ;UNRECOGNIZED; ERROR!
JMP      RDCODE        ;RETURN FOR MORE INPUT

```

```

;ROUTINE FOR HANDLING "MESSAGE OF THE DAY"
;HOST MAY SEND UP TO 80-CHARACTER MESSAGE TO SLAVE TO
;BE DISPLAYED ON THE VOLTMETER

```

```

MOV      *MESARG,R5
MOV      *6412,*FC
JSR      PC,WHITE
JMP      RDCODE

```

```

409
410
411
412 001762 013700 000000C
413 001762 162700 000002
414 001766 012701 0000003
415 001772 012701 0000003
416 001776 005033
417
418 002000 005002
419 002002 112102
420 002004 162702 000060
421 002010 000003
422 002012 005300
423 002014 003403
424 002016 070327
425 002022 000766
426 002024 000207
427
428
429
430
431
432
433 002026
434 002026 010046
435 002030 010146
436 002032 170200
437 002034 170300
438 002036 000900
439 002040 042700 100000
440 002044 170100
441 002046 012601
442 002050 012600
443 002052 000002
444 002054
445 002054 000000
446 002056 000002

;THIS SECTION WILL DECODE THE BYTE STRING FOLLOWING A
;SETUP COMMAND INTO AN INTEGER STORED IN R3.

DECODE:
MOV @1BLEN,R0
SUB #2,R0
MOV #1P,R1
CLR R3
;LENGTH OF RECEIVED STRING
;LESS CODE
;ADDRESS OF RECEIVED STRING
;CLEAR ACCUMULATED DATA WORD

CLR R2
MOV (R1)+,R2
SUB #60,R2
ADD R2,R3
DEC R0
BLE 20$
MUL #10,R3
BR 10$
RTS PC
;TO DE SAFE
;GET AN ASCII BYTE
;MAKE IT A BINARY DIGIT
;AND ADD TO DATA WORD
;DONE YET?
;FINISHED
;'SHIFT' LEFT
;NEXT DIGIT

10$:
20$:

;RUDIMENTARY HANDLING OF FLOATING-POINT EXCEPTIONS
;1. E. HALT ON ERROR; CLEAN UP ON CONTINUE (PROCEED)
;FLOATING-POINT INTERRUPT SERVICE

FPISR:
MOV R0,-(SP)
MOV R1,-(SP)
STFPS R0
STST R0
HALT
BIC #100000,R0
LDFPS R0
MOV (SP)+,R1
MOV (SP)+,R0
RTI

FOISR:
HALT
RTI

;ERROR INTERRUPTS FROM (FORTRAN) FUNCTIONS
;EG 0ATAN2, 00SIN, 0SQRT

```



```

505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542

;THIS PORTION WILL TYPE OUT ERROR MESSAGES:
;CONDITION? UNRECOGNIZED CODE.
      MOV     #CERRA,R5
      JSR     PC,WRITE
      MOV     #CERRC,R5
      JSR     PC,WRITE
      JNP     RDCODE
      2,IC,IBLEN,DLUN
      2,CERRT,CERRL,DLUN
      .ASCII  <12><15>"CODE ERROR: "
      .EVEN
      .WORD   14.

;THIS PORTION WILL HANDLE IRREGULAR INTERRUPTIONS,
;IE, PROCESS ABORTS FROM THE HOST PROCESSOR.
;ACTIVATED BY DEVICE TRIGGER (GET) ON BUS.
;ENTERED FROM GPIOSR / INTERRUPT SERVICE ROUTINE
      MOV     #STACK,SP
      MOV     #GPINA,R3
      JSR     PC,GPINI
      MOV     #ABO,R5
      JSR     PC,WRITE
      MTPS    R0
      JMP     RDCODE
      2,ABMES,ABLEN,DLUN
      .ASCII  <15>"*** FUNCTION ABORTED ***"
      .EVEN
      .WORD   25.

```

```

544      THIS PORTION OF THE PROGRAM CONTROLS THE A/D CONVERSION
545      AND DMA LOADING OF MEMORY WITH DATA.
546
547      CHECK POINTS REQUESTED AGAINST BUFFER SIZE.
548      TRY A/D CONVERSION AND REPEAT IF ERROR OCCURS.
549      INCREMENT CONVERSION (AND ERROR) COUNTER.
550
551      MEASURE:
552      002512      00000000      00000000      00000000
553      002512      00000000      00000000      00000000
554      002522      112737      177677      00000000
555      002530      000402
556
557      002532      004767      000022      1000
558
559      002536      012705      002552      00000000
560      002542      004767      00000000
561      002546      000167      176262
562      002552      000002      00000000      010164      GPACK: 2,1SW,TWO
563
564      002560      013737      00000000      00000000      00000000      00000000
565      002566      012705      002636      2000
566      002572      004767      00000000
567      002576      005237      00000000
568      002602      005737      00000000
569      002606      002403
570      002610      004767      004122
571      002614      000207
572      002616      005237      00000000      3000
573      002622      004767      004110
574      002626      005337      00000000
575      002632      003355
576      002634      000207
577
578      002636      000006      00000000      00000000      ADCOM: 6,ADBUF,GAIN,PAD,ADS,CATED,ADIC,1SW
579      002644      00000000      00000000      00000000
580      002652      00000000      00000000
581
582
583
584      002656
585      002656      005037      00000000
586      002662      005037      00000000
587      002666      004767      004044
588      002672      000167      176136
589

```

(THIS PORTION OF THE PROGRAM CONTROLS THE A/D CONVERSION
 AND DMA LOADING OF MEMORY WITH DATA.

 CHECK POINTS REQUESTED AGAINST BUFFER SIZE.
 TRY A/D CONVERSION AND REPEAT IF ERROR OCCURS.
 INCREMENT CONVERSION (AND ERROR) COUNTER.

 (LIMIT BUFFER TO MAX POINTS EACH CHANNEL
 WITHIN LIMIT??
 (NO--PARAMETER ERROR (-10))
 (ACKNOWLEDGE THIS
 (DO A/D CONVERSION
 (ACKNOWLEDGEMENT
 (NEXT COMMAND?

 (EXECUTE A/D CONVERSION
 (INCREMENT CONVERSION COUNTER
 (ANY ERRORS IN A/D?

 (NO ERRORS!
 (INCREMENT ERROR COUNTER
 (AND PRINT COUNT
 (ERROR, SO TRY AGAIN
 (BUT ONLY AS ALLOWED

 (THIS PORTION OF THE PROGRAM INITIALIZES THE TWO MAIN
 COUNTERS TO ZERO AND PRINTS THE "WAKE UP TEXT"

 (CLEAR A/D CONVERSION COUNTER
 (CLEAR A/D ERROR COUNTER
 (DISPLAY TEXT

THIS PORTION OF THE PROGRAM WILL LIST THE MAIN SETUP
PARAMETERS ON THE LSIVM CONSOLE.

```

591
592
593
594 002676 012737 000025 000000C PARLST:
595 002676 012737 000001 000000C MOV #21.,@LCTR
596 002704 012737 000000C MOV #1.,@LPTH
597 002712 012737 000000C MOV #DATA,@LADD
598 002720 012705 003042' IC$: MOV #LST1,R5
599 002724 004767 005240 JSR PC WRITE
600 002730 012705 003022' MOV #LSTNO,R5
601 002734 004767 005320 JSR PC PRINT
602 002740 012705 003064' MOV #LST2,R5
603 002744 004767 005220 JSR PC WRITE
604 002750 012705 003032' MOV #LSTDA,R5
605 002754 017737 000000C MOV @LADD,@LDDAT
606 002762 004767 005272 JSR PC PRINT
607 002766 005237 000000C INC @LADD
608 002772 005237 000000C INC @LPTH
609 002776 005237 000000C DEC @LCTR
610 003002 005337 000000C BLE 20$
611 003006 003403 JSR PC IFWAIT
612 003010 004767 000066 BR 10$
613 003014 000741
614
615 003016 000167 176012 20$: JMP RDCODE
616
617 003022 000002 000000C LSTNO: 2.LPTR,2,DLUN
618 003030 000000C 000000C LSTDA: 2.LDAT,6,DLUN
619 003032 000002 000000C 000000C LST1: 2.LST1,LSTL1,DLUN
620 003042 000002 003052' 003062' LST1: .ASCII <15><12>"DATA ("
621 003050 000000C 003052 015 012 104 LIST1: .EVEN
622 003055 015 124 101 .WORD LSTL1-LIST1
623 003060 040 050 2.LIST2,LSTL2,DLUN
624 003062 000010 LSTL1: .EVEN
625 003064 000002 003074' 003100' LST2: 2.LIST2,LSTL2,DLUN
626 003072 000000C 003074 051 040 075 LIST2: .EVEN
627 003077 040 040 LSTL2-LIST2
628 003100 000004 IFWAIT:
629 003102 012700 100000 10$: MOV #100000,R0
630 003106 005300 DEC R0
631 003110 000240 NOP
632 003112 000240 NOP
633 003114 000240 NOP
634 003116 000240 NOP
635 003120 001372 BNE 10$
636 003122 000207 RTS PC

```

```

639
640
641 003124
642 003124 012705 003140'
643 003124 004767 000000C
644 003130
645
646 003134 000167 175674
647 003140 000002 000000C 003146'
648 003146 000052
649
650
651
652
653
654
655
656
657 003150
658 003150 013737 003356' 000004C
659 003156 023727 000000C 053522
660 003164 001015
661 003166 013702 000000C
662 003172 005202
663 003174 006302
664 003176 006302
665 003200 010237 000000C
666 003204 012705 003350'
667 003210 004767 000000C
668 003214 000167 175614
669
670 003220 012701 000000C
671 003224 023727 000000C 030522
672 003232 001422
673 003234 023727 000000C 031122
674 003242 001412
675 003244 023727 000000C 031522
676 003252 001402 177044
677 003254 000167
678 003260 063701 000000C
679 003264 063701 000000C
680 003270 063701 000000C
681 003274 063701 000000C
682 003300 063701 000000C
683 003304 063701 000000C
684 003310 010137 000002C
685
686 003314 013702 000000C
687 003320 060202
688 003322 010237 000000C
689 003326 012737 000002 000000C
690 003334 012705 000000C
691 003340 004767 000000C
692
693 003344 000167 175464
694 003350 000002 000000C
695 003356 000000C

;THIS PORTION OF THE PROGRAM IS DESIGNED TO SEND THE DATA
;PARAMETERS BACK TO THE HOST PROCESSOR FOR EXAMINATION.

MOV $PARA,R5
JSR PC,CPOUT
;OUTPUT DATA

JMP RDCODE
;RETURN FOR MORE INPUT

2,DATA,PSIZE
WORD 42.

;THIS PORTION OF THE PROGRAM IS DESIGNED TO SEND ONE OF THE
;THREE DATA BUFFERS BACK TO THE HOST PROCESSOR. THE INPUT
;CODES ARE R1 (BUFFER 1), R2 (BUFFER 2), R3 (BUFFER 3).
;THE BUFFERS CONTAIN 12-BIT BINARY WORDS.

;ALSO TO TRANSFER BACK THE WINDOW (RW)---FLOATING WORDS.
;(PRECEDED BY COMPUTED AVERAGE).

ADSEND:
MOV $RANA,$RAA+4
CMP $R1C,$RW
BNE 5$
MOV $PTR,R2
INC R2
ASL R2
ASL R2
MOV $ADWL
MOV $ADWA,R5
JSR PC,CPOUT
JMP RDCODE
;OUTPUTTED
;RETURN FOR MORE INPUT

STORE ADDRESS
;WHAT CODE? (BUFFER?)

;ADD IN OFFSETS FOR
;...FOR BUFFERS 2 AND 3

;AND FOR THE STARTING POINT
;TWICE---WORD ADDRESS!

R1,$RAA+2

$PTR,R2
R2,R2
MOV R2,$RAN
MOV $2,$RAA
MOV $RAA,R5
JSR PC,CPOUT
;OUTPUT TO THE BUS

;RETURN FOR MORE INPUT
RDCODE
2,$AV,ADWL
RANA:

```


747					! COMPUTE AS SPECIFIED....
748					! RETURN DATA SPECIFIED....
749					
750	403:	003570	004767	001146	JSR PC,CSL
751		003574	004767	001256	JSR PC,TDFT
752		003600	000167	175230	JHP RDCODE
753					
754	503:	003604	004767	001054	JSR PC,CDFT
755		003610	004767	001242	JSR PC,TDFT
756		003614	000167	175214	JHP RDCODE
757					
758	603:	003620	004767	000764	JSR PC,CDFTD
759		003624	004767	001204	JSR PC,TDFTD
760		003630	000167	175200	JHP RDCODE
761					
762	703:	003634	004767	001342	JSR PC,GRNSV
763		003640	004767	001314	JSR PC,TRNSV
764		003644	000167	175164	JHP RDCODE
765					
766	803:	003650	004767	001420	JSR PC,CPEKO
767		003654	004767	001544	JSR PC,TPEK
768		003660	000167	175150	JHP RDCODE
769					
770	903:	003664	004767	001644	JSR PC,CPWR
771		003670	004767	002020	JSR PC,TPWR
772		003674	000167	175134	JHP RDCODE
773					
774	1003:	003700	004767	002030	JSR PC,CJC
775		003704	004767	002174	JSR PC,TJC
776		003710	000167	175120	JHP RDCODE
777					

```

778 ;ROUTINE TO COMPUTE A SINUSOIDAL WAVEFORM FOR TESTING.
779 ;V(1)=GAIN(N)*COS((1-1)*2*PI*NCYC/PTR)*2048./10.
780 ;TEST ROUTINE == NOT NECESSARILY NEAT OR EFFICIENT!
781
782 003714 012700 0000000C
783 003714 063700 0000000C
784 003720 063700 0000000C
785 003724 063700 0000000C
786 003730 063700 0000000C
787 003734 063700 0000000C
788 003740 013701 0000000C
789 003744 063701 0000000C
790 003750 060001 0000000C
791 003752 013702 0000000C
792 003756 063702 0000000C
793 003762 060102
794
795 003764 170001
796 003766 17237 0000000C
797 003772 17137 0000000C
798 003776 174601
799 004000 171237 004112
800
801 004004 013703 0000000C
802 004010 177003
803 004012 172067 000100
804 004016 171002
805 004020 010046
806 004022 010146
807 004024 010246
808 004026 010346
809 004030 174246
810 004032 004767 0000000C
811 004036 174437 004272
812 004042 172626
813 004044 012603
814 004046 012602
815 004050 012601
816 004052 012600
817
818 004054 177137 0000000C
819 004060 171100
820 004062 175540
821
822 004064 177137 0000000C
823 004070 171100
824 004072 175541
825
826 004074 177137 0000000C
827 004100 171100
828 004102 175542
829
830 004104 077337
831 004106 000167 174722
832 004112 040711 007733
833 004116 140200 000000
834

```

CWAIVE:
 MOV @ADBUF,R0
 ADD @STPT,R0
 ADD @STPT,R0
 ADD @PTR,R0
 ADD @PTR,R0
 MOV @PAD,R1
 ADD @PAD,R1
 ADD R0,R1
 MOV @PAD,R2
 ADD @PAD,R2
 ADD R1,R2

 SETF @NCYC,AC2
 LDCIF @PTR,AC1
 LDCIF AC1,AC2
 DIVF @TWOPI,AC2
 MULF

 MOV @PTR,R3
 R3,AC0
 ADDF N1,AC0
 MULF AC2,AC0
 MOV R0,-(SP)
 MOV R1,-(SP)
 MOV R2,-(SP)
 MOV R3,-(SP)
 STF AC2,-(SP)
 JSR PC,\$SCOS
 DIVF @FFAC,AC0
 LDF (SP)+,R3
 MOV (SP)+,R2
 MOV (SP)+,R1
 MOV (SP)+,R0

 LDCIF @GAIN1,AC1
 MULF AC0,AC1
 STCF1 AC1,-(R0)

 LDCIF @GAIN2,AC1
 MULF AC0,AC1
 STCF1 AC1,-(R1)

 LDCIF @GAIN3,AC1
 MULF AC0,AC1
 STCF1 AC1,-(R2)

 SOB R3,100
 JMP RDCODE
 TWOPI: .FLT2 6.2831854
 M1: .FLT2 -1.

;START OF FIRST BUFFER
 ;TRUE START OF FIRST WAVEFORM
 ;(ADD STARTING POINT--TWICE FOR BYTES)
 ;TRUE END OF FIRST WAVEFORM
 ;(ALSO TWICE)
 ;FIND END OF NEXT WAVEFORM
 ;BY ADDING TWICE POINTS TO A/D
 ;TRUE END OF SECOND WAVEFORM
 ;FIND END OF LAST WAVEFORM
 ;BY ADDING TWICE POINTS TO A/D
 ;TRUE END OF THIRD WAVEFORM

;NCYC FLOATED
 ;PTR FLOATED
 ;NCYC/PTR
 ;2*PI*(NCYC/PTR)

 ;POINTS TO COMPUTE
 ;FLOAT 1
 ;(1-1)
 ;(1-1)*(--)
 ;SAVE REGISTERS ON STACK

 ;COS(2*PI*(1-1)*NCYC/PTR)
 ;(--)*2048./10.
 ;RESTORE REGISTERS FROM STACK

;FLOAT GAIN (SCALE)
 ;GAIN(1)*(--)
 ;SIMULATED A/D CONVERTED VALUE

 ;AGAIN

 ;AGAIN

 ;NEXT SAMPLE (WORKING FROM END TO FRONT)

```

836 ROUTINE TO "FLOAT" THE A/D DATA BUFFER FOR PROCESSING
837 ; CONVERT INTEGER TO FLOATING NUMBERS IN VOLTS IN RBUF.
838
839 004122      MOV     R0,-(SP)      ;SAVE REGISTERS ON STACK
840 004122      MOV     R1,-(SP)
841 004124      MOV     R2,-(SP)
842 004126      SETF
843 004130      MOV     #ADBUF,R1   ;STORE ADDRESS
844 004132      CNPB     @1CN,'1     ;STARTING WHERE?
845 004136      BEQ     30$
846 004144      CNPD     @1CN,'2
847 004146      BEQ     20$
848 004154      CNPB     @1CN,'3
849 004156      BEQ     10$
850 004164      JNP     CERR
851 004166      ADD     @PAD,R1     ;ADD IN OFFSETS FOR
852 004172      ADD     @PAD,R1     ;...FOR BUFFERS 2 AND 3
853 004176      ADD     @PAD,R1
854 004202      ADD     @PAD,R1
855 004206      ADD     @STPT,R1   ;AND FOR THE STARTING POINT
856 004212      ADD     @STPT,R1   ;TWICE FOR BYTES
857 004216      MOV     @PTC,R0
858 004222      MOV     @RBUF,R2
859 004226      BIC     #170000,(R1) ;CLEAR TO TWELVE BITS
860 004232      CMP     (R1),#4000  ;IS IT NEGATIVE?
861 004236      BLT     110$
862 004242      BIS     #170000,(R1) ;YES--SIGN EXTEND
863 004244      LDCIF   (R1)+,AC0   ;CONVERT TO FLOATING
864 004250      MULF   @FFAC,AC0   ;DIVIDE BY HALF FULL SCALE (11 BITS)
865 004252      STF     AC0,(R2)+   ;MULTIPLY BY HALF FULL SCALE (10V)
866 004256      SOB     R0,100$
867 004262      MOV     (SP)+,R2
868 004264      MOV     (SP)+,R1
869 004266      MOV     (SP)+,R0
870 004270      RTS     PC
871 004272      .FLT2   .0048828125 ; = 19./2048.
872
873 004262      .FFAC:   000000
874 004264
875 004266
876 004270
877
878 004272      036240 000000

```

```

880 ROUTINE TO READ THE VALUES OF A SEQUENCE-WINDOW
881 FROM THE HOST.
882 ;NOTE THAT THE WINDOW MUST MATCH THE ENTIRE SEQUENCE
883 ;LENGTH (PTR=PTC).
884
885 004276      012765 004344'      WINDOW
886 004276      001767 00000000
887 004302
888
889
890
891
892
893 004306      170409
894 004310      012700 00000000
895 004314      011701 00000000
896 004320      170001
897 004322      172020
898 004324      077102
899 004326      177137
900 004332      174401
901 004334      173037 00000000
902
903 004340      003167 174470
904 004344      000002 004354'  PTR=
905 004352      00000000
906 004354      00000000
907
908
909
910
911
912 004356
913 004356
914 004362      001001
915 004364      000207
916
917 004366      012700 00000000
918 004372      013701 00000000
919 004376      012702 00000000
920
921 004402      170001
922 004404      172410
923 004406      172522
924 004410      171001
925 004412      174020
926 004414      077105
927
928 004416      000207

```

;DATA THAT HAS A WINDOW APPLIED IS REDUCED IN MAGNITUDE
 ;BY THE AVERAGE OF THE WINDOW. THIS AVERAGE MUST BE
 ;COMPUTED AND FACTORED INTO ALL RESULTS (VIA DECAIN).

;INITIALIZE SUM
 ;ADDRESS OF WINDOW DATA
 ;POINTS TO AVERAGE

;SUM WINDOW
 ;POINTS USED
 ;AVERAGED VALUE
 ;STAGED

;SUBROUTINE TO MULTIPLY THE DATA BUFFER DEUF BY THE
 ;STORED WINDOW FUNCTION.

;DESIREDT
 ;YES

;BUFFER ADDRESS
 ;POINTS TO BE WINDOWED
 ;WINDOW ADDRESS

;DATA WORD
 ;WINDOW WORD
 ;PRODUCT
 ;STORE BACK TO DATA ARRAY
 ;CONTINUE TILL DONE

;SUBROUTINE TO AVERAGE A SEQUENCE OF CYCLES---
 ;NOTE THAT IF THE NUMBER OF POINTS PER SUBSEQUENCE IS INTEGRAL,
 ;THE NCYC CYCLES CAN BE AVERAGED INTO NCYF CYCLE'S WORTH
 ;OF DATA, MAKING SUBSEQUENT COMPUTATION LESS TIME-CONSUMING.

930	004420	005737	000000C	HAVER:	TST	@AVGCCD	ALLOWABLE?
931	004420	001001			BNE	5\$	YES
932	004424	000207			RTS	PC	
933	004426	000207			CHP	#1,@NCYC	ONLY ONE CYCLE?
934	004430	001001	000000C 5\$:		BNE	10\$	NO
	004440	000207			RTS	PC	YES--FORGET IT!
942	004442	013701	000000C	10\$:	MOV	@NCYC,R1	NUMBER OF CYCLES PER SEQUENCE
943	004446	006700			SXT	R0	
944	004450	071037	000000C		DIV	@AVGCCD,R0	NUMBER OF CYCLES PER SUBSEQUENCE
945	004454	005701			TST	R1	COME OUT EVEN?
946	004456	001401			BEQ	13\$	OKAY
947	004460	000207			RTS	PC	NO--CAN'T AVERAGE
949	004462	022700	0000001	13\$:	CHP	#1,R0	NUMBER OF SUBSEQUENCES TO AVERAGE
950	004466	001001			BNE	14\$	ONE?
951	004470	000207			RTS	PC	YES--DON'T BOTHER
952	004472	010037	000000C	14\$:	MOV	R0,@NCAV	MORE--SAVE
954	004476	013701	000000C		MOV	@PTC,R1	TOTAL POINTS IN SEQUENCE
955	004502	070137	000000C		MUL	@AVGCCD,R1	..TIMES CYCLES IN SUBSEQUENCE
956	004506	006700			SXT	R0	
957	004510	071037	000000C		DIV	@NCYC,R0	DIVIDED BY CYCLES IN SEQUENCE
958	004514	005701			TST	R1	REMAINDER ZERO?
959	004516	001401			BEQ	20\$	YES; GO ON
960	004520	000207			RTS	PC	NO--CAN'T AVERAGE
961	004522	013737	000000C 20\$:		MOV	@AVGCCD,@NCYF	CYCLES PER SUBSEQUENCE; NON-ZERO
962	004530	010004			MOV	R0,R4	R0 = POINTS PER NEW SUBSEQUENCE
963	004532	010037	000000C		MOV	R0,@PTC	R4 = POINTS PER SUBSEQUENCE (COUNTER)
964	004536	006300			ASL	R0	NEW POINTS TO CONVERT
965	004540	006300	000000C		MOV	@NCAV,R1	R0 = ADDRESS OFFSET FOR 4-BYTE WORDS
966	004542	013701			MOV	@RBUF,R2	R1 = NUMBER OF SUBSEQUENCES (COUNTER)
967	004546	012702	000000C		MOV	@RBUF,R3	R2 = DATA ADDRESS (INPUT)
968	004552	012703	000000C		SETF	R1,AC1	R3 = DATA ADDRESS (OUTPUT)
969	004556	170001			LJCF	AC0	AC1 = NUMBER OF SUBSEQUENCES
970	004560	177101			CLRF	AC0	AC0 = RUNNING SUM
971	004562	170400	25\$:		CLRF	AC0	
972	004564	172012	30\$:		ADDF	(R2),AC0	SUM VALUES FROM EACH CYCLE
973	004566	060002			ADD	R0,R2	POINT TO NEXT CYCLE DATA WORD
974	004570	077103			SOB	R1,30\$	DONE YET?
975	004572	174401			DIVF	AC1,AC0	YES--COMPUTE AVERAGE
976	004574	174023			STF	AC0,(R3)+	SAVE AVERAGED POINT
977	004576	010302			MOV	R3,R2	NEXT INPUT ADDRESS
978	004578	013701	000000C		MOV	@NCAV,R1	RESTORE COUNTER
979	004580	077412			SOB	R4,25\$	NEXT POINT ON COUNTER
980	004600	000207			RTS	PC	

```

988      ;COMPUTE THE VOLTAGES VIA DISCRETE FOURIER TRANSFORMS (DFT):
989      ;TYPE "B"---VOLTAGE AND PHASE OF NCTH HARMONIC;
990      ;TYPE "C"---VOLTAGE AND PHASE OF FUNDAMENTAL;
991      ;TYPE "D"---VOLTAGE, PHASE, AND DISTORTION OF FUNDAMENTAL.
992      ;VOLTAGES ARE CORRECTED FOR A/D GAIN AND ARE RMS.
993      ;DISTORTION IS IN PERCENTAGE.
994
995      GDTF:
996      MOV      @ANAG,R5      ;COMPUTE THE DFT VALUES AND DISTORTION
997      JSR      PC,DAPD      ;DCRTZL-TYPE
998      LDF      @*VDFT,AC0    ;CORRECT FOR A/D GAIN
999      JSR      PC,DEGAIN    ;AND FOR WINDOW IF USED
1000      JSR      PC,DEWIND
1001      STF      AC0,@*VDFT
1002      RTS      PC
1003      7,RBUF,PTC,NCYF,VDFT,PDFT,DIST,HAR,IEND
1004
1005      GDTF:
1006      MOV      @ADFT,R5
1007      JSR      PC,DAPD
1008      LDF      @*VDFT,AC0
1009      JSR      PC,DEGAIN
1010      JSR      PC,DEWIND
1011      STF      AC0,@*VDFT
1012      RTS      PC
1013      7,RBUF,PTC,NCYF,VDFT,PDFT,DIST,NONE,IEND
1014
1015      .WORD 0
1016
1017      GSL:
1018      MOV      @*HTC,R5
1019      INC      R5
1020      MUL      @*NCYF,R5
1021      MOV      R5,@*NCYHA
1022
1023      MOV      @*ASL,R5
1024      JSR      PC,DAPD
1025      LDF      @*VDFT,AC0
1026      JSR      PC,DEGAIN
1027      JSR      PC,DEWIND
1028      STF      AC0,@*VDFT
1029      RTS      PC
1030
1031      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1032
1033      ASL:
1034      MOV      @*ASL,R5
1035      INC      R5
1036      MUL      @*NCYF,R5
1037      MOV      R5,@*NCYHA
1038
1039      MOV      @*ASL,R5
1040      JSR      PC,DAPD
1041      LDF      @*VDFT,AC0
1042      JSR      PC,DEGAIN
1043      JSR      PC,DEWIND
1044      STF      AC0,@*VDFT
1045      RTS      PC
1046
1047      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1048
1049      ASL:
1050      MOV      @*ASL,R5
1051      INC      R5
1052      MUL      @*NCYF,R5
1053      MOV      R5,@*NCYHA
1054
1055      MOV      @*ASL,R5
1056      JSR      PC,DAPD
1057      LDF      @*VDFT,AC0
1058      JSR      PC,DEGAIN
1059      JSR      PC,DEWIND
1060      STF      AC0,@*VDFT
1061      RTS      PC
1062
1063      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1064
1065      ASL:
1066      MOV      @*ASL,R5
1067      INC      R5
1068      MUL      @*NCYF,R5
1069      MOV      R5,@*NCYHA
1070
1071      MOV      @*ASL,R5
1072      JSR      PC,DAPD
1073      LDF      @*VDFT,AC0
1074      JSR      PC,DEGAIN
1075      JSR      PC,DEWIND
1076      STF      AC0,@*VDFT
1077      RTS      PC
1078
1079      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1080
1081      ASL:
1082      MOV      @*ASL,R5
1083      INC      R5
1084      MUL      @*NCYF,R5
1085      MOV      R5,@*NCYHA
1086
1087      MOV      @*ASL,R5
1088      JSR      PC,DAPD
1089      LDF      @*VDFT,AC0
1090      JSR      PC,DEGAIN
1091      JSR      PC,DEWIND
1092      STF      AC0,@*VDFT
1093      RTS      PC
1094
1095      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1096
1097      ASL:
1098      MOV      @*ASL,R5
1099      INC      R5
1100      MUL      @*NCYF,R5
1101      MOV      R5,@*NCYHA
1102
1103      MOV      @*ASL,R5
1104      JSR      PC,DAPD
1105      LDF      @*VDFT,AC0
1106      JSR      PC,DEGAIN
1107      JSR      PC,DEWIND
1108      STF      AC0,@*VDFT
1109      RTS      PC
1110
1111      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1112
1113      ASL:
1114      MOV      @*ASL,R5
1115      INC      R5
1116      MUL      @*NCYF,R5
1117      MOV      R5,@*NCYHA
1118
1119      MOV      @*ASL,R5
1120      JSR      PC,DAPD
1121      LDF      @*VDFT,AC0
1122      JSR      PC,DEGAIN
1123      JSR      PC,DEWIND
1124      STF      AC0,@*VDFT
1125      RTS      PC
1126
1127      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1128
1129      ASL:
1130      MOV      @*ASL,R5
1131      INC      R5
1132      MUL      @*NCYF,R5
1133      MOV      R5,@*NCYHA
1134
1135      MOV      @*ASL,R5
1136      JSR      PC,DAPD
1137      LDF      @*VDFT,AC0
1138      JSR      PC,DEGAIN
1139      JSR      PC,DEWIND
1140      STF      AC0,@*VDFT
1141      RTS      PC
1142
1143      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1144
1145      ASL:
1146      MOV      @*ASL,R5
1147      INC      R5
1148      MUL      @*NCYF,R5
1149      MOV      R5,@*NCYHA
1150
1151      MOV      @*ASL,R5
1152      JSR      PC,DAPD
1153      LDF      @*VDFT,AC0
1154      JSR      PC,DEGAIN
1155      JSR      PC,DEWIND
1156      STF      AC0,@*VDFT
1157      RTS      PC
1158
1159      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1160
1161      ASL:
1162      MOV      @*ASL,R5
1163      INC      R5
1164      MUL      @*NCYF,R5
1165      MOV      R5,@*NCYHA
1166
1167      MOV      @*ASL,R5
1168      JSR      PC,DAPD
1169      LDF      @*VDFT,AC0
1170      JSR      PC,DEGAIN
1171      JSR      PC,DEWIND
1172      STF      AC0,@*VDFT
1173      RTS      PC
1174
1175      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1176
1177      ASL:
1178      MOV      @*ASL,R5
1179      INC      R5
1180      MUL      @*NCYF,R5
1181      MOV      R5,@*NCYHA
1182
1183      MOV      @*ASL,R5
1184      JSR      PC,DAPD
1185      LDF      @*VDFT,AC0
1186      JSR      PC,DEGAIN
1187      JSR      PC,DEWIND
1188      STF      AC0,@*VDFT
1189      RTS      PC
1190
1191      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1192
1193      ASL:
1194      MOV      @*ASL,R5
1195      INC      R5
1196      MUL      @*NCYF,R5
1197      MOV      R5,@*NCYHA
1198
1199      MOV      @*ASL,R5
1200      JSR      PC,DAPD
1201      LDF      @*VDFT,AC0
1202      JSR      PC,DEGAIN
1203      JSR      PC,DEWIND
1204      STF      AC0,@*VDFT
1205      RTS      PC
1206
1207      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1208
1209      ASL:
1210      MOV      @*ASL,R5
1211      INC      R5
1212      MUL      @*NCYF,R5
1213      MOV      R5,@*NCYHA
1214
1215      MOV      @*ASL,R5
1216      JSR      PC,DAPD
1217      LDF      @*VDFT,AC0
1218      JSR      PC,DEGAIN
1219      JSR      PC,DEWIND
1220      STF      AC0,@*VDFT
1221      RTS      PC
1222
1223      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1224
1225      ASL:
1226      MOV      @*ASL,R5
1227      INC      R5
1228      MUL      @*NCYF,R5
1229      MOV      R5,@*NCYHA
1230
1231      MOV      @*ASL,R5
1232      JSR      PC,DAPD
1233      LDF      @*VDFT,AC0
1234      JSR      PC,DEGAIN
1235      JSR      PC,DEWIND
1236      STF      AC0,@*VDFT
1237      RTS      PC
1238
1239      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1240
1241      ASL:
1242      MOV      @*ASL,R5
1243      INC      R5
1244      MUL      @*NCYF,R5
1245      MOV      R5,@*NCYHA
1246
1247      MOV      @*ASL,R5
1248      JSR      PC,DAPD
1249      LDF      @*VDFT,AC0
1250      JSR      PC,DEGAIN
1251      JSR      PC,DEWIND
1252      STF      AC0,@*VDFT
1253      RTS      PC
1254
1255      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1256
1257      ASL:
1258      MOV      @*ASL,R5
1259      INC      R5
1260      MUL      @*NCYF,R5
1261      MOV      R5,@*NCYHA
1262
1263      MOV      @*ASL,R5
1264      JSR      PC,DAPD
1265      LDF      @*VDFT,AC0
1266      JSR      PC,DEGAIN
1267      JSR      PC,DEWIND
1268      STF      AC0,@*VDFT
1269      RTS      PC
1270
1271      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1272
1273      ASL:
1274      MOV      @*ASL,R5
1275      INC      R5
1276      MUL      @*NCYF,R5
1277      MOV      R5,@*NCYHA
1278
1279      MOV      @*ASL,R5
1280      JSR      PC,DAPD
1281      LDF      @*VDFT,AC0
1282      JSR      PC,DEGAIN
1283      JSR      PC,DEWIND
1284      STF      AC0,@*VDFT
1285      RTS      PC
1286
1287      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1288
1289      ASL:
1290      MOV      @*ASL,R5
1291      INC      R5
1292      MUL      @*NCYF,R5
1293      MOV      R5,@*NCYHA
1294
1295      MOV      @*ASL,R5
1296      JSR      PC,DAPD
1297      LDF      @*VDFT,AC0
1298      JSR      PC,DEGAIN
1299      JSR      PC,DEWIND
1300      STF      AC0,@*VDFT
1301      RTS      PC
1302
1303      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1304
1305      ASL:
1306      MOV      @*ASL,R5
1307      INC      R5
1308      MUL      @*NCYF,R5
1309      MOV      R5,@*NCYHA
1310
1311      MOV      @*ASL,R5
1312      JSR      PC,DAPD
1313      LDF      @*VDFT,AC0
1314      JSR      PC,DEGAIN
1315      JSR      PC,DEWIND
1316      STF      AC0,@*VDFT
1317      RTS      PC
1318
1319      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1320
1321      ASL:
1322      MOV      @*ASL,R5
1323      INC      R5
1324      MUL      @*NCYF,R5
1325      MOV      R5,@*NCYHA
1326
1327      MOV      @*ASL,R5
1328      JSR      PC,DAPD
1329      LDF      @*VDFT,AC0
1330      JSR      PC,DEGAIN
1331      JSR      PC,DEWIND
1332      STF      AC0,@*VDFT
1333      RTS      PC
1334
1335      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1336
1337      ASL:
1338      MOV      @*ASL,R5
1339      INC      R5
1340      MUL      @*NCYF,R5
1341      MOV      R5,@*NCYHA
1342
1343      MOV      @*ASL,R5
1344      JSR      PC,DAPD
1345      LDF      @*VDFT,AC0
1346      JSR      PC,DEGAIN
1347      JSR      PC,DEWIND
1348      STF      AC0,@*VDFT
1349      RTS      PC
1350
1351      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1352
1353      ASL:
1354      MOV      @*ASL,R5
1355      INC      R5
1356      MUL      @*NCYF,R5
1357      MOV      R5,@*NCYHA
1358
1359      MOV      @*ASL,R5
1360      JSR      PC,DAPD
1361      LDF      @*VDFT,AC0
1362      JSR      PC,DEGAIN
1363      JSR      PC,DEWIND
1364      STF      AC0,@*VDFT
1365      RTS      PC
1366
1367      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1368
1369      ASL:
1370      MOV      @*ASL,R5
1371      INC      R5
1372      MUL      @*NCYF,R5
1373      MOV      R5,@*NCYHA
1374
1375      MOV      @*ASL,R5
1376      JSR      PC,DAPD
1377      LDF      @*VDFT,AC0
1378      JSR      PC,DEGAIN
1379      JSR      PC,DEWIND
1380      STF      AC0,@*VDFT
1381      RTS      PC
1382
1383      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1384
1385      ASL:
1386      MOV      @*ASL,R5
1387      INC      R5
1388      MUL      @*NCYF,R5
1389      MOV      R5,@*NCYHA
1390
1391      MOV      @*ASL,R5
1392      JSR      PC,DAPD
1393      LDF      @*VDFT,AC0
1394      JSR      PC,DEGAIN
1395      JSR      PC,DEWIND
1396      STF      AC0,@*VDFT
1397      RTS      PC
1398
1399      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1400
1401      ASL:
1402      MOV      @*ASL,R5
1403      INC      R5
1404      MUL      @*NCYF,R5
1405      MOV      R5,@*NCYHA
1406
1407      MOV      @*ASL,R5
1408      JSR      PC,DAPD
1409      LDF      @*VDFT,AC0
1410      JSR      PC,DEGAIN
1411      JSR      PC,DEWIND
1412      STF      AC0,@*VDFT
1413      RTS      PC
1414
1415      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1416
1417      ASL:
1418      MOV      @*ASL,R5
1419      INC      R5
1420      MUL      @*NCYF,R5
1421      MOV      R5,@*NCYHA
1422
1423      MOV      @*ASL,R5
1424      JSR      PC,DAPD
1425      LDF      @*VDFT,AC0
1426      JSR      PC,DEGAIN
1427      JSR      PC,DEWIND
1428      STF      AC0,@*VDFT
1429      RTS      PC
1430
1431      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1432
1433      ASL:
1434      MOV      @*ASL,R5
1435      INC      R5
1436      MUL      @*NCYF,R5
1437      MOV      R5,@*NCYHA
1438
1439      MOV      @*ASL,R5
1440      JSR      PC,DAPD
1441      LDF      @*VDFT,AC0
1442      JSR      PC,DEGAIN
1443      JSR      PC,DEWIND
1444      STF      AC0,@*VDFT
1445      RTS      PC
1446
1447      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1448
1449      ASL:
1450      MOV      @*ASL,R5
1451      INC      R5
1452      MUL      @*NCYF,R5
1453      MOV      R5,@*NCYHA
1454
1455      MOV      @*ASL,R5
1456      JSR      PC,DAPD
1457      LDF      @*VDFT,AC0
1458      JSR      PC,DEGAIN
1459      JSR      PC,DEWIND
1460      STF      AC0,@*VDFT
1461      RTS      PC
1462
1463      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1464
1465      ASL:
1466      MOV      @*ASL,R5
1467      INC      R5
1468      MUL      @*NCYF,R5
1469      MOV      R5,@*NCYHA
1470
1471      MOV      @*ASL,R5
1472      JSR      PC,DAPD
1473      LDF      @*VDFT,AC0
1474      JSR      PC,DEGAIN
1475      JSR      PC,DEWIND
1476      STF      AC0,@*VDFT
1477      RTS      PC
1478
1479      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1480
1481      ASL:
1482      MOV      @*ASL,R5
1483      INC      R5
1484      MUL      @*NCYF,R5
1485      MOV      R5,@*NCYHA
1486
1487      MOV      @*ASL,R5
1488      JSR      PC,DAPD
1489      LDF      @*VDFT,AC0
1490      JSR      PC,DEGAIN
1491      JSR      PC,DEWIND
1492      STF      AC0,@*VDFT
1493      RTS      PC
1494
1495      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1496
1497      ASL:
1498      MOV      @*ASL,R5
1499      INC      R5
1500      MUL      @*NCYF,R5
1501      MOV      R5,@*NCYHA
1502
1503      MOV      @*ASL,R5
1504      JSR      PC,DAPD
1505      LDF      @*VDFT,AC0
1506      JSR      PC,DEGAIN
1507      JSR      PC,DEWIND
1508      STF      AC0,@*VDFT
1509      RTS      PC
1510
1511      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1512
1513      ASL:
1514      MOV      @*ASL,R5
1515      INC      R5
1516      MUL      @*NCYF,R5
1517      MOV      R5,@*NCYHA
1518
1519      MOV      @*ASL,R5
1520      JSR      PC,DAPD
1521      LDF      @*VDFT,AC0
1522      JSR      PC,DEGAIN
1523      JSR      PC,DEWIND
1524      STF      AC0,@*VDFT
1525      RTS      PC
1526
1527      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1528
1529      ASL:
1530      MOV      @*ASL,R5
1531      INC      R5
1532      MUL      @*NCYF,R5
1533      MOV      R5,@*NCYHA
1534
1535      MOV      @*ASL,R5
1536      JSR      PC,DAPD
1537      LDF      @*VDFT,AC0
1538      JSR      PC,DEGAIN
1539      JSR      PC,DEWIND
1540      STF      AC0,@*VDFT
1541      RTS      PC
1542
1543      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1544
1545      ASL:
1546      MOV      @*ASL,R5
1547      INC      R5
1548      MUL      @*NCYF,R5
1549      MOV      R5,@*NCYHA
1550
1551      MOV      @*ASL,R5
1552      JSR      PC,DAPD
1553      LDF      @*VDFT,AC0
1554      JSR      PC,DEGAIN
1555      JSR      PC,DEWIND
1556      STF      AC0,@*VDFT
1557      RTS      PC
1558
1559      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1560
1561      ASL:
1562      MOV      @*ASL,R5
1563      INC      R5
1564      MUL      @*NCYF,R5
1565      MOV      R5,@*NCYHA
1566
1567      MOV      @*ASL,R5
1568      JSR      PC,DAPD
1569      LDF      @*VDFT,AC0
1570      JSR      PC,DEGAIN
1571      JSR      PC,DEWIND
1572      STF      AC0,@*VDFT
1573      RTS      PC
1574
1575      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1576
1577      ASL:
1578      MOV      @*ASL,R5
1579      INC      R5
1580      MUL      @*NCYF,R5
1581      MOV      R5,@*NCYHA
1582
1583      MOV      @*ASL,R5
1584      JSR      PC,DAPD
1585      LDF      @*VDFT,AC0
1586      JSR      PC,DEGAIN
1587      JSR      PC,DEWIND
1588      STF      AC0,@*VDFT
1589      RTS      PC
1590
1591      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1592
1593      ASL:
1594      MOV      @*ASL,R5
1595      INC      R5
1596      MUL      @*NCYF,R5
1597      MOV      R5,@*NCYHA
1598
1599      MOV      @*ASL,R5
1600      JSR      PC,DAPD
1601      LDF      @*VDFT,AC0
1602      JSR      PC,DEGAIN
1603      JSR      PC,DEWIND
1604      STF      AC0,@*VDFT
1605      RTS      PC
1606
1607      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1608
1609      ASL:
1610      MOV      @*ASL,R5
1611      INC      R5
1612      MUL      @*NCYF,R5
1613      MOV      R5,@*NCYHA
1614
1615      MOV      @*ASL,R5
1616      JSR      PC,DAPD
1617      LDF      @*VDFT,AC0
1618      JSR      PC,DEGAIN
1619      JSR      PC,DEWIND
1620      STF      AC0,@*VDFT
1621      RTS      PC
1622
1623      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1624
1625      ASL:
1626      MOV      @*ASL,R5
1627      INC      R5
1628      MUL      @*NCYF,R5
1629      MOV      R5,@*NCYHA
1630
1631      MOV      @*ASL,R5
1632      JSR      PC,DAPD
1633      LDF      @*VDFT,AC0
1634      JSR      PC,DEGAIN
1635      JSR      PC,DEWIND
1636      STF      AC0,@*VDFT
1637      RTS      PC
1638
1639      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1640
1641      ASL:
1642      MOV      @*ASL,R5
1643      INC      R5
1644      MUL      @*NCYF,R5
1645      MOV      R5,@*NCYHA
1646
1647      MOV      @*ASL,R5
1648      JSR      PC,DAPD
1649      LDF      @*VDFT,AC0
1650      JSR      PC,DEGAIN
1651      JSR      PC,DEWIND
1652      STF      AC0,@*VDFT
1653      RTS      PC
1654
1655      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1656
1657      ASL:
1658      MOV      @*ASL,R5
1659      INC      R5
1660      MUL      @*NCYF,R5
1661      MOV      R5,@*NCYHA
1662
1663      MOV      @*ASL,R5
1664      JSR      PC,DAPD
1665      LDF      @*VDFT,AC0
1666      JSR      PC,DEGAIN
1667      JSR      PC,DEWIND
1668      STF      AC0,@*VDFT
1669      RTS      PC
1670
1671      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1672
1673      ASL:
1674      MOV      @*ASL,R5
1675      INC      R5
1676      MUL      @*NCYF,R5
1677      MOV      R5,@*NCYHA
1678
1679      MOV      @*ASL,R5
1680      JSR      PC,DAPD
1681      LDF      @*VDFT,AC0
1682      JSR      PC,DEGAIN
1683      JSR      PC,DEWIND
1684      STF      AC0,@*VDFT
1685      RTS      PC
1686
1687      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1688
1689      ASL:
1690      MOV      @*ASL,R5
1691      INC      R5
1692      MUL      @*NCYF,R5
1693      MOV      R5,@*NCYHA
1694
1695      MOV      @*ASL,R5
1696      JSR      PC,DAPD
1697      LDF      @*VDFT,AC0
1698      JSR      PC,DEGAIN
1699      JSR      PC,DEWIND
1700      STF      AC0,@*VDFT
1701      RTS      PC
1702
1703      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1704
1705      ASL:
1706      MOV      @*ASL,R5
1707      INC      R5
1708      MUL      @*NCYF,R5
1709      MOV      R5,@*NCYHA
1710
1711      MOV      @*ASL,R5
1712      JSR      PC,DAPD
1713      LDF      @*VDFT,AC0
1714      JSR      PC,DEGAIN
1715      JSR      PC,DEWIND
1716      STF      AC0,@*VDFT
1717      RTS      PC
1718
1719      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1720
1721      ASL:
1722      MOV      @*ASL,R5
1723      INC      R5
1724      MUL      @*NCYF,R5
1725      MOV      R5,@*NCYHA
1726
1727      MOV      @*ASL,R5
1728      JSR      PC,DAPD
1729      LDF      @*VDFT,AC0
1730      JSR      PC,DEGAIN
1731      JSR      PC,DEWIND
1732      STF      AC0,@*VDFT
1733      RTS      PC
1734
1735      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1736
1737      ASL:
1738      MOV      @*ASL,R5
1739      INC      R5
1740      MUL      @*NCYF,R5
1741      MOV      R5,@*NCYHA
1742
1743      MOV      @*ASL,R5
1744      JSR      PC,DAPD
1745      LDF      @*VDFT,AC0
1746      JSR      PC,DEGAIN
1747      JSR      PC,DEWIND
1748      STF      AC0,@*VDFT
1749      RTS      PC
1750
1751      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1752
1753      ASL:
1754      MOV      @*ASL,R5
1755      INC      R5
1756      MUL      @*NCYF,R5
1757      MOV      R5,@*NCYHA
1758
1759      MOV      @*ASL,R5
1760      JSR      PC,DAPD
1761      LDF      @*VDFT,AC0
1762      JSR      PC,DEGAIN
1763      JSR      PC,DEWIND
1764      STF      AC0,@*VDFT
1765      RTS      PC
1766
1767      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1768
1769      ASL:
1770      MOV      @*ASL,R5
1771      INC      R5
1772      MUL      @*NCYF,R5
1773      MOV      R5,@*NCYHA
1774
1775      MOV      @*ASL,R5
1776      JSR      PC,DAPD
1777      LDF      @*VDFT,AC0
1778      JSR      PC,DEGAIN
1779      JSR      PC,DEWIND
1780      STF      AC0,@*VDFT
1781      RTS      PC
1782
1783      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1784
1785      ASL:
1786      MOV      @*ASL,R5
1787      INC      R5
1788      MUL      @*NCYF,R5
1789      MOV      R5,@*NCYHA
1790
1791      MOV      @*ASL,R5
1792      JSR      PC,DAPD
1793      LDF      @*VDFT,AC0
1794      JSR      PC,DEGAIN
1795      JSR      PC,DEWIND
1796      STF      AC0,@*VDFT
1797      RTS      PC
1798
1799      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1800
1801      ASL:
1802      MOV      @*ASL,R5
1803      INC      R5
1804      MUL      @*NCYF,R5
1805      MOV      R5,@*NCYHA
1806
1807      MOV      @*ASL,R5
1808      JSR      PC,DAPD
1809      LDF      @*VDFT,AC0
1810      JSR      PC,DEGAIN
1811      JSR      PC,DEWIND
1812      STF      AC0,@*VDFT
1813      RTS      PC
1814
1815      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1816
1817      ASL:
1818      MOV      @*ASL,R5
1819      INC      R5
1820      MUL      @*NCYF,R5
1821      MOV      R5,@*NCYHA
1822
1823      MOV      @*ASL,R5
1824      JSR      PC,DAPD
1825      LDF      @*VDFT,AC0
1826      JSR      PC,DEGAIN
1827      JSR      PC,DEWIND
1828      STF      AC0,@*VDFT
1829      RTS      PC
1830
1831      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1832
1833      ASL:
1834      MOV      @*ASL,R5
1835      INC      R5
1836      MUL      @*NCYF,R5
1837      MOV      R5,@*NCYHA
1838
1839      MOV      @*ASL,R5
1840      JSR      PC,DAPD
1841      LDF      @*VDFT,AC0
1842      JSR      PC,DEGAIN
1843      JSR      PC,DEWIND
1844      STF      AC0,@*VDFT
1845      RTS      PC
1846
1847      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1848
1849      ASL:
1850      MOV      @*ASL,R5
1851      INC      R5
1852      MUL      @*NCYF,R5
1853      MOV      R5,@*NCYHA
1854
1855      MOV      @*ASL,R5
1856      JSR      PC,DAPD
1857      LDF      @*VDFT,AC0
1858      JSR      PC,DEGAIN
1859      JSR      PC,DEWIND
1860      STF      AC0,@*VDFT
1861      RTS      PC
1862
1863      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1864
1865      ASL:
1866      MOV      @*ASL,R5
1867      INC      R5
1868      MUL      @*NCYF,R5
1869      MOV      R5,@*NCYHA
1870
1871      MOV      @*ASL,R5
1872      JSR      PC,DAPD
1873      LDF      @*VDFT,AC0
1874      JSR      PC,DEGAIN
1875      JSR      PC,DEWIND
1876      STF      AC0,@*VDFT
1877      RTS      PC
1878
1879      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1880
1881      ASL:
1882      MOV      @*ASL,R5
1883      INC      R5
1884      MUL      @*NCYF,R5
1885      MOV      R5,@*NCYHA
1886
1887      MOV      @*ASL,R5
1888      JSR      PC,DAPD
1889      LDF      @*VDFT,AC0
1890      JSR      PC,DEGAIN
1891      JSR      PC,DEWIND
1892      STF      AC0,@*VDFT
1893      RTS      PC
1894
1895      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1896
1897      ASL:
1898      MOV      @*ASL,R5
1899      INC      R5
1900      MUL      @*NCYF,R5
1901      MOV      R5,@*NCYHA
1902
1903      MOV      @*ASL,R5
1904      JSR      PC,DAPD
1905      LDF      @*VDFT,AC0
1906      JSR      PC,DEGAIN
1907      JSR      PC,DEWIND
1908      STF      AC0,@*VDFT
1909      RTS      PC
1910
1911      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1912
1913      ASL:
1914      MOV      @*ASL,R5
1915      INC      R5
1916      MUL      @*NCYF,R5
1917      MOV      R5,@*NCYHA
1918
1919      MOV      @*ASL,R5
1920      JSR      PC,DAPD
1921      LDF      @*VDFT,AC0
1922      JSR      PC,DEGAIN
1923      JSR      PC,DEWIND
1924      STF      AC0,@*VDFT
1925      RTS      PC
1926
1927      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1928
1929      ASL:
1930      MOV      @*ASL,R5
1931      INC      R5
1932      MUL      @*NCYF,R5
1933      MOV      R5,@*NCYHA
1934
1935      MOV      @*ASL,R5
1936      JSR      PC,DAPD
1937      LDF      @*VDFT,AC0
1938      JSR      PC,DEGAIN
1939      JSR      PC,DEWIND
1940      STF      AC0,@*VDFT
1941      RTS      PC
1942
1943      7,RBUF,PTC,NCYHA,VDFT,PDFT,DIST,NONE,IEND
1944
1945      ASL:
1946      MOV      @*ASL,R5
1947      INC      R5
1948      MUL      @*NCYF,R5
1949      MOV      R5,@*NCYHA
1950
1951      MOV      @*ASL,R5
1952      JSR      PC,DAPD
1953      LDF      @*VDFT,AC0
1954      JSR      PC,DEGAIN
1955      JSR      PC,DEWIND
1956      STF      AC0,@*VDFT
1957      RTS      PC
1958
1959      7
```

```

1033
1034
1035
1036 005034
1037 005034 012705 005046'
1038 005040 004767 000000G
1039 005044 000415
1040 005046 000002 000000G 005054'
1041 005054 000014
1042
1043 005056
1044 005056 012705 005070'
1045 005062 004767 000000G
1046 005066 000413
1047 005070 000002 000000G 005076'
1048 005076 000010
1049
1050 005100 172437 000000G
1051 005104 012700 000000G
1052 005110 063700 000000G
1053 005114 174010
1054 005116 172437 000000G
1055 005122 012700 000000G
1056 005126 063700 000000G
1057 005132 174010
1058 005134 172437 000000G
1059 005140 012700 000000G
1060 005144 063700 000000G
1061 005150 174010
1062 005152 172437 000000G
1063 005156 172537 000000G
1064 005162 173001
1065 005164 174037 000000G
1066
1067 005170 004767 001752
1068 005174 004767 002364
1069 005200 000207

;ROUTINE TO RETURN DATA TO THE HOST.
;ALSO TO SET UP ON-LINE DISPLAY VALUES.

TDFTD:
MOV 0TDFTA,R5
JSR PC,CPOUT
BR DSLD
2.DFD,N12
.WORD 12.

;OUTPUT VOLTAGE, PHASE, DISTORTION

TDFTA:
MOV 0TDFTA,R5
JSR PC,CPOUT
BR DSLD
2.DFD,NB
.WORD 8.

;OUTPUT VOLTAGE, PHASE

TDFT:
MOV 0TDFTA,R5
JSR PC,CPOUT
BR DSLD
2.DFD,NB
.WORD 8.

;LOAD DATA FOR DISPLAY ROUTINES
;BASE ADDRESS FOR DISTORTIONS
;PARTICULAR ONE AVAILABLE
;LOAD IT
;SAME FOR VOLTAGE

DSL:
LDF 0DIST,AC0
MOV 0DIST1,R0
ADD 0CHNO,R0
STF AC0,(R0)
LDF 0VDFT,AC0
MOV 0RHSV1,R0
ADD 0CHNO,R0
STF AC0,(R0)
LDF 0PDFT,AC0
MOV 0PHAS1,R0
ADD 0CHNO,R0
STF AC0,(R0)
LDF 0PHAS1,AC0
LDF 0PHAS2,AC1
SUBF AC1,AC0
STF AC0,0RPHAS

;SAME FOR PHASE

;COMPUTE RELATIVE PHASE

;POSSIBLE---ONE CHANNEL DISPLAY
;POSSIBLE---TWO CHANNEL/RELATIVE PHASE

PC,CHIDIS
PC,CH5DIS
RTS

```

```

1071      ;COMPUTE THE VOLTAGE PARAMETERS BY RMS TIME-AVERAGE:
1072      ;TYPE "E" -- GET RMS, PEAK, AND DC VOLTAGES.
1073      ;TYPE "F" -- GET ABS(PEAK) VOLTAGE ONLY.
1074      ;VOLTAGES ARE CORRECTED FOR A/D CONVERTER GAIN.
1075
1076      GRMSV:
1077      MOV 005202 012705 005342'
1078      JSR 005206 004767 000000C
1079
1080      LDF 005212 172437 000000C
1081      JSR 005216 004767 000246
1082      STF 005222 174037 000000C
1083      LDF 005226 172437 000000C
1084      JSR 005232 004767 000232
1085      STF 005236 174037 000000C
1086      LDF 005242 172437 000000C
1087      JSR 005246 004767 000216
1088      STF 005252 174037 000000C
1089      LDF 005256 172437 000000C
1090      JSR 005262 004767 000202
1091      STF 005266 174037 000000C
1092      RTS 005272 000207
1093
1094      GPEKO:
1095      MOV 005274 012701 000000C
1096      JSR 005300 013700 000000C
1097      LDF 005304 172421
1098      ABSF 005306 170600
1099      DEC 005310 005300
1100      LDZ 005312 172521
1101      ABSF 005314 170601
1102      CHPF 005316 173401
1103      CFCC 005320 170000
1104      BLE 005322 003401
1105      LDF 005324 172401
1106      SOB 005326 077007
1107      JSR 005330 004767 000134
1108      STF 005334 174037 000000C
1109      RTS 005340 000207
1110
1111      ARMSV:
1112      MOV 005342 000006 000000C
1113      JSR 005350 000000C 000000C
1114      RTS 005356 000000C

```

```

1113
1114
1115
1116 005360 012705 005414'
1117 005360 004767 000000G
1118 005364 004767 000000G
1119 005370 172437 000000G
1120 005374 012700 000000G
1121 005400 063700 000000G
1122 005404 174010 002154
1123 005406 004767 000000G
1124 005412 000207 000000G
1125 005414 000002 000000G
1126 005422 000020 000000G
1127
1128 005424
1129 005424 012705 005460'
1130 005430 004767 000000G
1131 005434 172437 000000G
1132 005440 012700 000000G
1133 005444 063700 000000G
1134 005450 174010 002110
1135 005452 004767 000000G
1136 005456 000207 000000G
1137 005460 000002 000000G
1138 005466 000004 000000G
1139
1140
1141
1142
1143
1144
1145
1146
1147 005470
1148 005470 113700 000000G
1149 005474 042700 177774
1150 005500 060000 000000G
1151 005502 062700 177776G
1152 005506 170001 000000G
1153 005510 177110 000000G
1154
1155 005512 174401 000000G
1156 005514 000207 000000G
1157
1158 005516
1159 005516 005737 000000G
1160 005522 001403 000000G
1161 005524 172537 000000G
1162 005530 174401 000000G
1163 005532 000207 000000G

;SUBROUTINES TO TRANSMIT RMS DATA TO THE HOST.
;ALSO PREPARATION FOR REAL-TIME DISPLAYS.

MOV #TMSA,R5
JSR PC,GPOUT
LDF @RHSV,AC0
MOV #RHSV1,R0
ADD @CHNO,R0
STF AC0,(R0)
JSR PC,CH4DIS
RTS PC
2.RMD,N16
.NORD 16.

TMSV:
TMSA: 005422'
N16:

TPEK:
TPEKA: 005466'
N4:

;TRANSFER THE DATA BACK
;STORE FOR DISPLAYS
;BASE ADDRESS OF RMS DATA
;OFFSET FOR PARTICULAR CHANNEL

;TRANSFER PEAK VOLTAGE ONLY
;LOAD FOR DISPLAY USE
;BASE ADDRESS FOR "RMS" DATA
;OFFSET FOR PARTICULAR CHANNEL
;STORE PEAK "AS RMS"

;SUBROUTINE TO CONVERT THE COMPUTED VOLTAGES TO TRUE VALUES
;BY REMOVING THE GAIN OF THE A/D CONVERTERS.

;ALSO TO CORRECT DATA COMPUTED WITH A KAISER WINDOW, BY
;DIVIDING THE RESULTING MAGNITUDE BY THE WINDOW AVERAGE.

DECAIN:
MOV B @ICN,R0
BIC #177774,R0
ADD R0,R0
ADD #GAIN-2,R0
SETF (R0),AC1
LDCIF AC1,AC0
DIVF AC1,AC0
RTS PC

DEWIND:
TST @WMODE
REQ 100
LDF @NAV,AC1
DIVF AC1,AC0
RTS PC

;SUBROUTINE TO REMOVE WINDOW FACTOR
;USING A WINDOW?
;NOT THIS TIME
;GET AVERAGE (NAV)
;VDFT = VDFT / NAV

```

```

1165
1166
1167
1168
1169
1170 005534
1171 005534 010046
1172 005536 010146
1173 005540 010246
1174 005542 170001
1175 005544 012701
1176 005550 000000G
1177 005554 000000G
1178 005560 010102
1179 005562 000000G
1180 005566 000000G
1181 005572 013700
1182
1183 005576 170402
1184
1185
1186 005600 042711 170000
1187 005604 021127 004000
1188 005610 002402
1189 005612 052711 170000
1190 005616 177021
1191 005620 171037 004272
1192
1193 005624 177137 000000G
1194 005630 174401
1195
1196
1197 005632 042712 170000
1198 005636 021227 004000
1199 005642 002402
1200 005644 052712 170000
1201 005650 177122
1202 005652 171137 004272
1203
1204 005656 171001
1205 005660 177137 000000G
1206 005664 174401
1207 005666 172200
1208 005670 077035
1209
1210 005672 177037 000000G
1211 005676 174600
1212 005700 174237 000000G
1213 005704 012602
1214 005706 012601
1215 005710 012600
1216 005712 000207
1217
1218 005714 012705 005726
1219 005720 004767 000000G
1220 005724 000207
1221 005726 000002 000000G 005466

```

GPWR:

```

MOV R0,-(SP)
MOV R1,-(SP)
MOV R2,-(SP)
SETF
MOV @DBUF,R1
ADD @STPT,R1
ADD @STPT,R1
MOV R1,R2
ADD @PAD,R2
ADD @PAD,R2
MOV @PTR,R0
CLRf AC2

```

1000:

```

BIC #170000,(R1)
CMP (R1),#4000
BLT 1100
BIS #170000,(R1)
LDCIF (R1)+,AC0
MULF @FFAC,AC0

```

1100:

```

LDCIF
DIVF @GAIN1,AC1
AC1,AC0

```

1200:

```

BIC #170000,(R2)
CMP (R2),#4000
BLT 1300
BIS #170000,(R2)
LDCIF (R2)+,AC1
MULF @FFAC,AC1

```

1300:

```

MULF AC1,AC0
LDCIF @GAIN2,AC1
DIVF AC1,AC0
ADDF AC0,AC2
SOB R0,1000

```

LDCIF

```

DIVF AC0,AC2
STF AC2,@POWER
MOV (SP)+,R2
MOV (SP)+,R1
MOV (SP)+,R0
RTS PC

```

TPWR:

```

MOV #TPWRA,R5
JSR PC,CPOUT
RTS PC

```

TPWRA:

```

2,POWER,N4

```

;SUBROUTINES TO COMPUTE THE TRUE (PSEUDO) POWER
;ASSUMING VOLTAGE AND CURRENT ARE IN CHANNELS #1-#2
; PWR/K = SUM (V1(J)*V2(J)) / N <FOR J=1,N>
;PSEUDO, FOR TRUE CURRENT IS UNKNOWN. I(J) = K*V2(J).
;COMPUTE PSEUDO-POWER
;SAVE REGISTERS ON STACK
;STORE ADDRESS OF VOLTAGE
;INCLUDING STARTING POINT
;TIMES TWO (INTEGER ADDRESS)
;STORE ADDRESS OF CURRENT
;OFFSET TO SECOND BUFFER
;ALREADY INCLUDES STARTING POINT OFFSET
;AND POINTS TO COMPUTE
;INITIALIZE SUM
;FLOAT VOLTAGE.....
;CLEAR TO TWELVE BITS
;IS IT NEGATIVE?
;NO
;YES--SIGN EXTEND
;CONVERT TO FLOATING
;DIVIDE BY HALF FULL SCALE (11 BITS)
;MULTIPLY BY HALF FULL SCALE (10V)
;GET GAIN OF VOLTAGE CHANNEL
;CONVERT TO TRUE VOLTS
;FLOAT CURRENT.....
;CLEAR TO TWELVE BITS
;IS IT NEGATIVE?
;NO
;YES--SIGN EXTEND
;CONVERT TO FLOATING
;DIVIDE BY HALF FULL SCALE (11 BITS)
;MULTIPLY BY HALF FULL SCALE (10V)
;GET GAIN OF CURRENT CHANNEL
;CORRECT CURRENT FOR GAIN
;SUM PRODUCTS
;NEXT?
;POINTS DONE
;TAKE AVERAGE
;AND SAVE
;RESTORE REGISTERS FROM STACK
;RETURN PSEUDO-POWER


```

;VOLTMEETER SIMULATION ROUTINE.
;IN THIS STATE THE SLAVE PROCESSOR WILL PRETEND TO BE A
;SAMPLING VOLTMEETER AND MEASURE/COMPUTE/DISPLAY AS FAST
;AS IT CAN. PRIOR TO THIS MODE BEING ENTERED, VARIOUS
;PARAMETERS NEED TO BE SET UP (AS IN THE NORMAL COMPUTER-
;CONTROLLED SITUATION). THIS MODE PROCEEDS INDEPENDENT
;OF ANY HOST CONTROL, ALTHOUGH THE HOST CAN ABORT THE MODE
;AND THE SLAVE WILL REVERT TO THE INPUT-AWAITING MODE.

```

Address	Operation	Comments
1273	MOV	PC, DOAD
1274	MOV	PC, DOAD
1275	MOV	PC, DOAD
1276	MOV	PC, DOAD
1277	MOV	PC, DOAD
1278	MOV	PC, DOAD
1279	MOV	PC, DOAD
1280	MOV	PC, DOAD
1281	MOV	PC, DOAD
1282	MOV	PC, DOAD
1283	MOV	PC, DOAD
1284	MOV	PC, DOAD
1285	MOV	PC, DOAD
1286	MOV	PC, DOAD
1287	MOV	PC, DOAD
1288	MOV	PC, DOAD
1289	MOV	PC, DOAD
1290	MOV	PC, DOAD
1291	MOV	PC, DOAD
1292	MOV	PC, DOAD
1293	MOV	PC, DOAD
1294	MOV	PC, DOAD
1295	MOV	PC, DOAD
1296	MOV	PC, DOAD
1297	MOV	PC, DOAD
1298	MOV	PC, DOAD
1299	MOV	PC, DOAD
1300	MOV	PC, DOAD
1301	MOV	PC, DOAD
1302	MOV	PC, DOAD
1303	MOV	PC, DOAD
1304	MOV	PC, DOAD
1305	MOV	PC, DOAD
1306	MOV	PC, DOAD
1307	MOV	PC, DOAD
1308	MOV	PC, DOAD
1309	MOV	PC, DOAD
1310	MOV	PC, DOAD
1311	MOV	PC, DOAD
1312	MOV	PC, DOAD
1313	MOV	PC, DOAD
1314	MOV	PC, DOAD
1315	MOV	PC, DOAD
1316	MOV	PC, DOAD
1317	MOV	PC, DOAD
1318	MOV	PC, DOAD
1319	MOV	PC, DOAD
1320	MOV	PC, DOAD
1321	MOV	PC, DOAD
1322	MOV	PC, DOAD
1323	MOV	PC, DOAD
1324	MOV	PC, DOAD
1325	MOV	PC, DOAD
1326	MOV	PC, DOAD
1327	MOV	PC, DOAD
1328	MOV	PC, DOAD
1329	MOV	PC, DOAD
1330	MOV	PC, DOAD
1331	MOV	PC, DOAD
1332	MOV	PC, DOAD
1333	MOV	PC, DOAD
1334	MOV	PC, DOAD
1335	MOV	PC, DOAD
1336	MOV	PC, DOAD
1337	MOV	PC, DOAD
1338	MOV	PC, DOAD
1339	MOV	PC, DOAD
1340	MOV	PC, DOAD
1341	MOV	PC, DOAD
1342	MOV	PC, DOAD
1343	MOV	PC, DOAD
1344	MOV	PC, DOAD
1345	MOV	PC, DOAD
1346	MOV	PC, DOAD
1347	MOV	PC, DOAD
1348	MOV	PC, DOAD
1349	MOV	PC, DOAD
1350	MOV	PC, DOAD
1351	MOV	PC, DOAD
1352	MOV	PC, DOAD
1353	MOV	PC, DOAD
1354	MOV	PC, DOAD
1355	MOV	PC, DOAD
1356	MOV	PC, DOAD
1357	MOV	PC, DOAD
1358	MOV	PC, DOAD
1359	MOV	PC, DOAD
1360	MOV	PC, DOAD
1361	MOV	PC, DOAD
1362	MOV	PC, DOAD
1363	MOV	PC, DOAD
1364	MOV	PC, DOAD
1365	MOV	PC, DOAD
1366	MOV	PC, DOAD
1367	MOV	PC, DOAD
1368	MOV	PC, DOAD
1369	MOV	PC, DOAD
1370	MOV	PC, DOAD
1371	MOV	PC, DOAD
1372	MOV	PC, DOAD
1373	MOV	PC, DOAD
1374	MOV	PC, DOAD
1375	MOV	PC, DOAD
1376	MOV	PC, DOAD
1377	MOV	PC, DOAD
1378	MOV	PC, DOAD
1379	MOV	PC, DOAD
1380	MOV	PC, DOAD
1381	MOV	PC, DOAD
1382	MOV	PC, DOAD
1383	MOV	PC, DOAD
1384	MOV	PC, DOAD
1385	MOV	PC, DOAD
1386	MOV	PC, DOAD
1387	MOV	PC, DOAD
1388	MOV	PC, DOAD
1389	MOV	PC, DOAD
1390	MOV	PC, DOAD
1391	MOV	PC, DOAD
1392	MOV	PC, DOAD
1393	MOV	PC, DOAD
1394	MOV	PC, DOAD
1395	MOV	PC, DOAD
1396	MOV	PC, DOAD
1397	MOV	PC, DOAD
1398	MOV	PC, DOAD
1399	MOV	PC, DOAD
1400	MOV	PC, DOAD
1401	MOV	PC, DOAD
1402	MOV	PC, DOAD
1403	MOV	PC, DOAD
1404	MOV	PC, DOAD
1405	MOV	PC, DOAD
1406	MOV	PC, DOAD
1407	MOV	PC, DOAD
1408	MOV	PC, DOAD
1409	MOV	PC, DOAD
1410	MOV	PC, DOAD
1411	MOV	PC, DOAD
1412	MOV	PC,

```

1322      006370 112737 000061 0000000C 808:
1323      006376 004767 175320
1324      006376 004767 175320
1325      006402 004767 175750
1326      006406 004767 176006
1327      006412 004767 176564
1328      006416 172437 0000000C
1329      006422 174037 0000000C
1330      006426 112737 000062 0000000C
1331      006434 013737 0000000C 0000000C
1332      006442 013737 0000000C 0000000C
1333      006450 004767 175446
1334      006454 004767 175676
1335      006460 004767 175734
1336      006464 004767 176512
1337      006470 172437 0000000C
1338      006474 174037 0000000C
1339      006500 112737 000063 0000000C
1340      006506 013737 0000000C 0000000C
1341      006514 013737 0000000C 0000000C
1342      006522 004767 175374
1343      006526 004767 175624
1344      006532 004767 175662
1345      006536 004767 176440
1346      006542 172437 0000000C
1347      006546 174037 0000000C
1348      006552 004767 001010
1349      006556 000465
1350      BR
1351

1352      006560 112737 000061 0000000C 908:
1353      006566 004767 175330
1354      006572 004767 175560
1355      006576 004767 175616
1356      006602 004767 176056
1357      006606 172437 0000000C
1358      006612 174037 0000000C
1359      006616 172437 0000000C
1360      006622 174037 0000000C
1361      006626 112737 000062 0000000C
1362      006634 013737 0000000C 0000000C
1363      006642 013737 0000000C 0000000C
1364      006650 004767 175246
1365      006654 004767 175476
1366      006660 004767 175534
1367      006664 004767 175774
1368      006670 172437 0000000C
1369      006674 174037 0000000C
1370      006700 172437 0000000C
1371      006704 174037 0000000C
1372      006710 172437 0000000C
1373      006714 172537 0000000C
1374      006720 173001
1375      006722 174037 0000000C
1376      006726 004767 001032
1377
1378      006732 000167 177166
1379      JNP 1008:

```

```

;DISPLAY MODE 4 (THREE CHANNELS)
;CHANNEL 1
;FLOAT DATA
;MULTIPLY BY WINDOW
;AVERAGE DATA
;GET VOLTAGE
;ACCESS VOLTAGE

;CHANNEL 2
;POINTS TO COMPUTE ON
;CYCLES TO COMPUTE
;FLOAT DATA
;MULTIPLY BY WINDOW
;AVERAGE DATA
;GET VOLTAGE
;ACCESS VOLTAGE

;CHANNEL 3
;POINTS TO COMPUTE ON
;CYCLES TO COMPUTE
;FLOAT DATA
;MULTIPLY BY WINDOW
;AVERAGE DATA
;GET VOLTAGE
;ACCESS VOLTAGE

;DISPLAY DATA

;DISPLAY MODE 5 (DUAL CHANNEL)
;FLOAT DATA
;MULTIPLY BY WINDOW
;AVERAGE DATA
;GET VOLTAGE AND PHASE
;ACCESS VOLTAGE

;ACCESS PHASE

;CHANNEL 2
;POINTS TO COMPUTE ON
;CYCLES TO COMPUTE

;MULTIPLY BY WINDOW
;AVERAGE DATA
;GET VOLTAGE AND PHASE
;ACCESS VOLTAGE

;ACCESS PHASE

;COMPUTE RELATIVE PHASE

;DISPLAY DATA

;AND AGAIN

```

```

1380      006736 005737 000000G
1381      006742 001401
1382      006744 000207
1383      006746 012700 000000G
1384      006752 012701 007114'
1385      006756 112120
1386      006760 001376
1387      006762 013701 000000G
1388      006766 006700
1389      006770 012702 000005
1390      006774 012703 000017G
1391      006776 012707 000012
1392      006780 062701 000060
1393      006784 110143
1394      006788 010001
1395      006792 006700
1396      006796 007210
1397      006800 013701 000000G
1398      006804 006700
1399      006808 012702 000005
1400      006812 012703 000017G
1401      006816 012707 000012
1402      006820 062701 000060
1403      006824 110143
1404      006828 010001
1405      006832 006700
1406      006836 007210
1407      006840 013701 000000G
1408      006844 006700
1409      006848 012702 000005
1410      006852 012703 000017G
1411      006856 012707 000012
1412      006860 062701 000060
1413      006864 110143
1414      006868 010001
1415      006872 006700
1416      006876 007210
1417      006880 013701 000000G
1418      006884 006700
1419      006888 012702 000005
1420      006892 012703 000017G
1421      006896 012707 000012
1422      006900 062701 000060
1423      006904 110143
1424      006908 010001
1425      006912 006700
1426      006916 007210
1427      006920 013701 000000G
1428      006924 006700
1429      006928 012702 000005
1430      006932 012703 000017G
1431      006936 012707 000012
1432      006940 062701 000060
1433      006944 110143
1434      006948 010001
1435      006952 006700
1436      006956 007210
1437      006960 013701 000000G
1438      006964 006700
1439      006968 012702 000005
1440      006972 012703 000017G
1441      006976 012707 000012
1442      006980 062701 000060
1443      006984 110143
1444      006988 010001
1445      006992 006700
1446      006996 007210
1447      007000 013701 000000G
1448      007004 006700
1449      007008 012702 000005
1450      007012 012703 000017G
1451      007016 012707 000012
1452      007020 062701 000060
1453      007024 110143
1454      007028 010001
1455      007032 006700
1456      007036 007210
1457      007040 013701 000000G
1458      007044 006700
1459      007048 012702 000005
1460      007052 012703 000017G
1461      007056 012707 000012
1462      007060 062701 000060
1463      007064 110143
1464      007068 010001
1465      007072 006700
1466      007076 007210
1467      007080 013701 000000G
1468      007084 006700
1469      007088 012702 000005
1470      007092 012703 000017G
1471      007096 012707 000012
1472      007100 062701 000060
1473      007104 110143
1474      007108 010001
1475      007112 006700
1476      007116 007210
1477      007120 013701 000000G
1478      007124 006700
1479      007128 012702 000005
1480      007132 012703 000017G
1481      007136 012707 000012
1482      007140 062701 000060
1483      007144 110143
1484      007148 010001
1485      007152 006700
1486      007156 007210
1487      007160 013701 000000G
1488      007164 006700
1489      007168 012702 000005
1490      007172 012703 000017G
1491      007176 012707 000012
1492      007180 062701 000060
1493      007184 110143
1494      007188 010001
1495      007192 006700
1496      007196 007210
1497      007200 013701 000000G
1498      007204 006700
1499      007208 012702 000005
1500      007212 012703 000017G
1501      007216 012707 000012
1502      007220 062701 000060
1503      007224 110143
1504      007228 010001
1505      007232 006700
1506      007236 007210
1507      007240 013701 000000G
1508      007244 006700
1509      007248 012702 000005
1510      007252 012703 000017G
1511      007256 012707 000012
1512      007260 062701 000060
1513      007264 110143
1514      007268 010001
1515      007272 006700
1516      007276 007210
1517      007280 013701 000000G
1518      007284 006700
1519      007288 012702 000005
1520      007292 012703 000017G
1521      007296 012707 000012
1522      007300 062701 000060
1523      007304 110143
1524      007308 010001
1525      007312 006700
1526      007316 007210
1527      007320 013701 000000G
1528      007324 006700
1529      007328 012702 000005
1530      007332 012703 000017G
1531      007336 012707 000012
1532      007340 062701 000060
1533      007344 110143
1534      007348 010001
1535      007352 006700
1536      007356 007210
1537      007360 013701 000000G
1538      007364 006700
1539      007368 012702 000005
1540      007372 012703 000017G
1541      007376 012707 000012
1542      007380 062701 000060
1543      007384 110143
1544      007388 010001
1545      007392 006700
1546      007396 007210
1547      007400 013701 000000G
1548      007404 006700
1549      007408 012702 000005
1550      007412 012703 000017G
1551      007416 012707 000012
1552      007420 062701 000060
1553      007424 110143
1554      007428 010001
1555      007432 006700
1556      007436 007210
1557      007440 013701 000000G
1558      007444 006700
1559      007448 012702 000005
1560      007452 012703 000017G
1561      007456 012707 000012
1562      007460 062701 000060
1563      007464 110143
1564      007468 010001
1565      007472 006700
1566      007476 007210
1567      007480 013701 000000G
1568      007484 006700
1569      007488 012702 000005
1570      007492 012703 000017G
1571      007496 012707 000012
1572      007500 062701 000060
1573      007504 110143
1574      007508 010001
1575      007512 006700
1576      007516 007210
1577      007520 013701 000000G
1578      007524 006700
1579      007528 012702 000005
1580      007532 012703 000017G
1581      007536 012707 000012
1582      007540 062701 000060
1583      007544 110143
1584      007548 010001
1585      007552 006700
1586      007556 007210
1587      007560 013701 000000G
1588      007564 006700
1589      007568 012702 000005
1590      007572 012703 000017G
1591      007576 012707 000012
1592      007580 062701 000060
1593      007584 110143
1594      007588 010001
1595      007592 006700
1596      007596 007210
1597      007600 013701 000000G
1598      007604 006700
1599      007608 012702 000005
1600      007612 012703 000017G
1601      007616 012707 000012
1602      007620 062701 000060
1603      007624 110143
1604      007628 010001
1605      007632 006700
1606      007636 007210
1607      007640 013701 000000G
1608      007644 006700
1609      007648 012702 000005
1610      007652 012703 000017G
1611      007656 012707 000012
1612      007660 062701 000060
1613      007664 110143
1614      007668 010001
1615      007672 006700
1616      007676 007210
1617      007680 013701 000000G
1618      007684 006700
1619      007688 012702 000005
1620      007692 012703 000017G
1621      007696 012707 000012
1622      007700 062701 000060
1623      007704 110143
1624      007708 010001
1625      007712 006700
1626      007716 007210
1627      007720 013701 000000G
1628      007724 006700
1629      007728 012702 000005
1630      007732 012703 000017G
1631      007736 012707 000012
1632      007740 062701 000060
1633      007744 110143
1634      007748 010001
1635      007752 006700
1636      007756 007210
1637      007760 013701 000000G
1638      007764 006700
1639      007768 012702 000005
1640      007772 012703 000017G
1641      007776 012707 000012
1642      007780 062701 000060
1643      007784 110143
1644      007788 010001
1645      007792 006700
1646      007796 007210
1647      007800 013701 000000G
1648      007804 006700
1649      007808 012702 000005
1650      007812 012703 000017G
1651      007816 012707 000012
1652      007820 062701 000060
1653      007824 110143
1654      007828 010001
1655      007832 006700
1656      007836 007210
1657      007840 013701 000000G
1658      007844 006700
1659      007848 012702 000005
1660      007852 012703 000017G
1661      007856 012707 000012
1662      007860 062701 000060
1663      007864 110143
1664      007868 010001
1665      007872 006700
1666      007876 007210
1667      007880 013701 000000G
1668      007884 006700
1669      007888 012702 000005
1670      007892 012703 000017G
1671      007896 012707 000012
1672      007900 062701 000060
1673      007904 110143
1674      007908 010001
1675      007912 006700
1676      007916 007210
1677      007920 013701 000000G
1678      007924 006700
1679      007928 012702 000005
1680      007932 012703 000017G
1681      007936 012707 000012
1682      007940 062701 000060
1683      007944 110143
1684      007948 010001
1685      007952 006700
1686      007956 007210
1687      007960 013701 000000G
1688      007964 006700
1689      007968 012702 000005
1690      007972 012703 000017G
1691      007976 012707 000012
1692      007980 062701 000060
1693      007984 110143
1694      007988 010001
1695      007992 006700
1696      007996 007210
1697      008000 013701 000000G
1698      008004 006700
1699      008008 012702 000005
1700      008012 012703 000017G
1701      008016 012707 000012
1702      008020 062701 000060
1703      008024 110143
1704      008028 010001
1705      008032 006700
1706      008036 007210
1707      008040 013701 000000G
1708      008044 006700
1709      008048 012702 000005
1710      008052 012703 000017G
1711      008056 012707 000012
1712      008060 062701 000060
1713      008064 110143
1714      008068 010001
1715      008072 006700
1716      008076 007210
1717      008080 013701 000000G
1718      008084 006700
1719      008088 012702 000005
1720      008092 012703 000017G
1721      008096 012707 000012
1722      008100 062701 000060
1723      008104 110143
1724      008108 010001
1725      008112 006700
1726      008116 007210
1727      008120 013701 000000G
1728      008124 006700
1729      008128 012702 000005
1730      008132 012703 000017G
1731      008136 012707 000012
1732      008140 062701 000060
1733      008144 110143
1734      008148 010001
1735      008152 006700
1736      008156 007210
1737      008160 013701 000000G
1738      008164 006700
1739      008168 012702 000005
1740      008172 012703 000017G
1741      008176 012707 000012
1742      008180 062701 000060
1743      008184 110143
1744      008188 010001
1745      008192 006700
1746      008196 007210
1747      008200 013701 000000G
1748      008204 006700
1749      008208 012702 000005
1750      008212 012703 000017G
1751      008216 012707 000012
1752      008220 062701 000060
1753      008224 110143
1754      008228 010001
1755      008232 006700
1756      008236 007210
1757      008240 013701 000000G
1758      008244 006700
1759      008248 012702 000005
1760      008252 012703 000017G
1761      008256 012707 000012
1762      008260 062701 000060
1763      008264 110143
1764      008268 010001
1765      008272 006700
1766      008276 007210
1767      008280 013701 000000G
1768      008284 006700
1769      008288 012702 000005
1770      008292 012703 000017G
1771      008296 012707 000012
1772      008300 062701 000060
1773      008304 110143
1774      008308 010001
1775      008312 006700
1776      008316 007210
1777      008320 013701 000000G
1778      008324 006700
1779      008328 012702 000005
1780      008332 012703 000017G
1781      008336 012707 000012
1782      008340 062701 000060
1783      008344 110143
1784      008348 010001
1785      008352 006700
1786      008356 007210
1787      008360 013701 000000G
1788      008364 006700
1789      008368 012702 000005
1790      008372 012703 000017G
1791      008376 012707 000012
1792      008380 062701 000060
1793      008384 110143
1794      008388 010001
1795      008392 006700
1796      008396 007210
1797      008400 013701 000000G
1798      008404 006700
1799      008408 012702 000005
1800      008412 012703 000017G
1801      008416 012707 000012
1802      008420 062701 000060
1803      008424 110143
1804      008428 010001
1805      008432 006700
1806      008436 007210
1807      008440 013701 000000G
1808      008444 006700
1809      008448 012702 000005
1810      008452 012703 000017G
1811      008456 012707 000012
1812      008460 062701 000060
1813      008464 110143
1814      008468 010001
1815      008472 006700
1816      008476 007210
1817      008480 013701 000000G
1818      008484 006700
1819      008488 012702 000005
1820      008492 012703 000017G
1821      008496 012707 000012
1822      008500 062701 000060
1823      008504 110143
1824      008508 010001
1825      008512 006700
1826      008516 007210
1827      008520 013701 000000G
1828      008524 006700
1829      008528 012702 000005
1830      008532 012703 000017G
1831      008536 012707 000012
1832      008540 062701 000060
1833      008544 110143
1834      008548 010001
1835      008552 006700
1836      008556 007210
1837      008560 013701 000000G
1838      008564 006700
1839      008568 012702 000005
1840      008572 012703 000017G
1841      008576 012707 000012
1842      008580 062701 000060
1843      008584 110143
1844      008588 010001
1845      008592 006700
1846      008596 007210
1847      008600 013701 000000G
1848      008604 006700
1849      008608 012702 000005
1850      008612 012703 000017G
1851      008616 012707 000012
1852      008620 062701 000060
1853      008624 110143
1854      008628 010001
1855      008632 006700
1856      008636 007210
1857      008640 013701 000000G
1858      008644 006700
1859      008648 012702 000005
1860      008652 012703 000017G
1861      008656 012707 000012
1862      008660 062701 000060
1863      008664 110143
1864      008668 010001
1865      008672 006700
1866      008676 007210
1867      008680 013701 000000G
1868      008684 006700
1869      008688 012702 000005
1870      008692 012703 000017G
1871      008696 012707 000012
1872      008700 062701 000060
1873      008704 110143
1874      008708 010001
1875      008712 006700
1876      008716 007210
1877      008720 013701 000000G
1878      008724 006700
1879      008728 012702 000005
1880      008732 012703 000017G
1881      008736 012707 000012
1882      008740 062701 000060
1883      008744 110143
1884      008748 010001
1885      008752 006700
1886      008756 007210
1887      008760 013701 000000G
1888      008764 006700
1889      008768 012702 000005
1890      008772 012703 000017G
1891      008776 012707 000012
1892      008780 062701 000060
1893      008784 110143
1894      008788 010001
1895      008792 006700
1896      008796 007210
1897      008800 013701 000000G
1898      008804 006
```

```

1427
1428
1429
1430 007146
1431 007146
1432 007146
1433 007146
1434 007152
1435 007156
1436 007160
1437
1438 007162
1439 007170
1440 007172
1441 007176
1442 007202
1443 007206
1444 007212
1445 007216
1446 007222
1447 007230
1448
1449 007232
1450 007240
1451 007242
1452 007246
1453 007252
1454 007256
1455 007262
1456 007266
1457 007272
1458 007300
1459
1460 007302
1461 007310
1462 007312
1463 007316
1464 007322
1465 007326
1466 007332
1467 007336
1468 007342
1469 007350
1470
1471 007352
1472
1473 007354
1474 007360
1475 007364
1476

CH2DIS:
CH3DIS:
50:
012700 000000G
012701 007532'
112120
001376
023727 000000G 000001
001020
007366'
000000G
012705 007400'
004767 000000G
012705 007412'
012705 000000G
004767 000000G
000061 000001G
112737 000451
023727 000000G 000002 100:
001020
007424'
000000G
012705 007436'
004767 000000G
012705 007450'
004767 000000G
000062 000001G
000425
023727 000000G 000003 200:
001020
007462'
000000G
012705 007474'
004767 000000G
012705 007506'
004767 000000G
000063 000001G
000401
000207
012705 007520'
004767 000604
000207

```

SUBROUTINE TO DISPLAY DATA ON VOLTMETER
;FORMAT # (1-3)= >V=+***.*** P: +****.* D:*.**

```

;LOAD DISPLAY DELIMITERS
;INTO RAM BUFFER
50
;CHERRY,R0
;CH1BLK,R1
(R1)+,(R0)+
50
;DSTYLE,#1
100
;CH1A,R5
PC,CONVRT
;CH1B,R5
PC,CONVRT
;CH1C,R5
PC,CONVRT
;CH1,0=CHERRY+1
400
;? (CHANNEL 2)
200
;CH2A,R5
PC,CONVRT
;CH2B,R5
PC,CONVRT
;CH2C,R5
PC,CONVRT
;CH2,0=CHERRY+1
400
;DSTYLE,#3
300
;CH3A,R5
PC,CONVRT
;CH3B,R5
PC,CONVRT
;CH3C,R5
PC,CONVRT
;CH3,0=CHERRY+1
400
;NO
;VALUE FILL DISPLAY
;CH1R,R5
PC,WRITE
PC
RTS
MOV
JSR
RTS

```

1478 007366 000004 000000C 000005C CH1A: 4,RMSV1,CHERRY+5.,TWO,TWO
 007374 010164' 010164'
 1479 007400 000004 000000C 000016C CH1B: 4,PHAS1,CHERRY+14.,THREE,ONE
 007406 010166' 010162'
 1480 007412 000004 000000C 000027C CH1C: 4,DIST1,CHERRY+23.,ONE,TWO
 007420 010162' 010164'
 1481 007424 000004 000000C 000005C CH2A: 4,RMSV2,CHERRY+5.,TWO,TWO
 007432 010164' 010164'
 1482 007436 000004 000000C 000016C CH2B: 4,PHAS2,CHERRY+14.,THREE,ONE
 007444 010166' 010162'
 1483 007450 000004 000000C 000027C CH2C: 4,DIST2,CHERRY+23.,ONE,TWO
 007456 010162' 010164'
 1484 007462 000004 000000C 000005C CH3A: 4,RMSV3,CHERRY+5.,TWO,TWO
 007470 010164' 010164'
 1485 007474 000004 000000C 000016C CH3B: 4,PHAS3,CHERRY+14.,THREE,ONE
 007502 010166' 010162'
 1486 007506 000004 000000C 000027C CH3C: 4,DIST3,CHERRY+23.,ONE,TWO
 007514 010162' 010164'
 1487 007520 000002 000000C 007530' CHIER: 2,CHERRY,CHILA,DLUN

CHILA: .WORD 28.

1488 007530 000034
 1489
 1490 007532 015 061 075 075 053
 007535 126 075 053
 007540 043 043 056
 007543 043 043 040
 007546 120 075 053
 007551 043 043 043
 007554 056 043 040
 007557 104 075 043
 007562 056 043
 007565 000

1491 .EVEN

```

1493
1494
1495
1496 007556 023727 000000C 000004 CH4DIS:
1497 007556 001401
1498 007574 000207
1499 007576 000207
1500 007600 012700
1501 007604 012701
1502 007610 112120
1503 007612 001376
1504 007614 012705
1505 007620 004767
1506 007624 012705
1507 007630 004767
1508 007634 012705
1509 007640 004767
1510 007644 012705
1511 007650 004767
1512 007654 000207
1513
1514 007656 000004 000000C CH4A: 4.RMSV1,CHERRY+4.,ONE,TWO
1515 007670 000004 000000C CH4B: 4.RMSV2,CHERRY+13.,ONE,TWO
1516 007702 000004 000000C CH4C: 4.RMSV3,CHERRY+22.,ONE,THREE
1517 007714 000002 CH4ER: 2.CHERRY,CH4LA,DLUN
1518 007722 000034 CH4LA: .WORD 28.
1519
1520 007726 015 126 061 CH4BLK: .ASCIZ <15>*V1=+*,** V2=+*,** V3=+*,***
007731 075 053 043
007734 056 043 043
007737 040 126 062
007742 075 053 043
007745 056 043 043
007750 040 126 063
007753 075 053 043
007756 056 043 043
007761 043 000
1521

```

;SUBROUTINE TO DISPLAY DATA ON VOLTMETER
 ;FORMAT 4: V1=+*,** V2=+*,** V3=+*,***
 CMP @DSTYLE.,#4
 BEQ 10\$
 RTS PC
 RTS PC
 MOV #CHERRY,R0
 MOV #CH4BLK,R1
 MOV (R1)+,(R0)+
 BNE 20\$
 MOV #CH4A,R5
 PC,CONVRT
 MOV #CH4B,R5
 PC,CONVRT
 MOV #CH4C,R5
 PC,CONVRT
 JSR #CH4ER,R5
 MOV PC,WRITE
 JSR PC
 RTS PC
 ;RIGHT DISPLAY STYLE?
 ;YES
 ;NO
 ;LOAD DISPLAY DELIMITERS
 ;INTO RAM BUFFER
 ;CONVERT CHANNEL ONE VOLTAGE TO ASCII
 ;CONVERT CHANNEL TWO VOLTAGE TO ASCII
 ;CONVERT CHANNEL THREE VOLTAGE TO ASCII
 ;VALUE FULL DISPLAY

```

1523
1524
1525
1526 007764
1527 007764 023727 0000000C 0000005 CH5DIS:
1528 007772 001401 BEQ 100 000000C 0000005 CH5DIS:
1529 007774 000207 RTS PC
1530 007776 012700 0000000C 0000005 CH5DIS:
1531 010002 012701 010124' 100:
1532 010006 112120 200:
1533 010010 001376 BNE 200
1534 010012 012705 010054'
1535 010016 004767 0000000C
1536 010022 012705 010066'
1537 010026 004767 0000000C
1538 010032 012705 010100'
1539 010036 004767 0000000C
1540 010042 012705 010112'
1541 010046 004767 000116
1542 010052 000207
1543
1544 010054 000004 0000000C 000004C CH5A:
1545 010062 010162' 010166'
1546 010074 010162' 010166'
1547 010106 000004 0000000C 000027C CH5C:
1548 010112 000002 0000000C 010122' CH5ER:
1549 010120 0000000C
1550 010122 000034 CH5LA:
1551 010124 015 126 061
1552 010127 075 053 043
1553 010132 056 043 043
1554 010135 043 040 126
1555 010140 062 075 053
1556 010143 043 056 043
1557 010146 043 043 040
1558 010151 120 075 053
1559 010154 043 043 056
1560 010157 043 000
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000

```

```

1557      !SUBROUTINES TO HANDLE CHARACTER OUTPUT:
1558      !CALL WRITE (IBUFF,ILEN,DLUN)---ASCII STRING
1559      !CALL PRINT (NUMBER,ILEN,DLUN)---DECIMAL NUMBER
1560
1561      !SET UP START ADDRESS
1562      !SET UP LENGTH
1563      !IF NEG OR ZERO, ERROR
1564      !GET NEXT CHARACTER
1565      !LOOP ON CHAR COUNT
1566
1567      2(R5),R0
1568      @4(R5),R1
1569      30$
1570      MOVB (R0)+,R2
1571      PC,PUTCHR
1572      R1,10$
1573      RTS
1574
1575      !WHICH DEVICE?
1576      !CHERRY DISPLAY
1577      !CHECK XMIT STATUS
1578      !NOT DONE YET
1579      !MOVE IT TO PRINT BUF
1580
1581      *2,@6(R5)
1582      CHCHR
1583      *200,@DLVXSR
1584      PUTCHR
1585      R2,@DLVXBF
1586      PC
1587
1588      !CHECK XMIT STATUS
1589      !NOT DONE YET
1590      !MOVE IT TO PRINT BUF
1591
1592      *200,@CHXSR
1593      CHCHR
1594      R2,@CHXBF
1595      PC
1596
1597      !PRINT A DECIMAL INTEGER
1598
1599      !SETUP OUTPUT BUFFER
1600      !GET ROW MANY DIGITS
1601      !GET NUMBER TO PRINT
1602      !WORRY IF NEGATIVE
1603      !ONE LESS TO PRINT
1604      !AND EXTEND TO 32 BITS
1605
1606      *STRING,R3
1607      4(R5),R2
1608      @2(R5),R1
1609      R1
1610      5$
1611      BGE R1
1612      NEG R1
1613      DEC R2
1614      SXT R0
1615
1616      !DIVIDE BY 10
1617      !CONVERT REMAINDER TO ASCII
1618      !AND STORE IN STRING
1619      !CONVERT QUOTIENT TO DIVIDEND
1620      !AND TO 32 BITS
1621      !CHECK NUMBER OF DIGITS
1622      !GET ANOTHER
1623
1624      *10,R0
1625      *60,R1
1626      R1,(R3)+
1627      MOV R0,R1
1628      SXT R0
1629      DEC R2
1630      BGT 10$
1631
1632      !CHECK SIGN
1633      20$
1634      *55,R2
1635      PC,PUTCHR
1636
1637      !CHARACTERS TO PRINT
1638      !ADDRESS OF STRING (MSD)
1639      !PRINT DIGIT
1640      !DONE YET?
1641      !NO
1642      !YES
1643
1644      4(R5),R1
1645      -(R3),R2
1646      PC,PUTCHR
1647      R1
1648      30$
1649      BGT 30$
1650      RTS
1651
1652      .END

```

(2) MODULE A32D - Subroutine to control ADAC A/D converters

Calling Sequence:

CALL A32D (BUFFER, GAINS, NWORDS, ADS, GATED, CHNO, ISW)

where the arguments are:

BUFFER	Integer array to receive contiguous sampled data
GAINS	Integer array containing gains of individual A/D converters
NWORDS	Number of samples to digitize per channel
ADS	A/D setup word containing which A/D's in use; which to interrupt from
GATED	Integer variable specifying presence (1) or absence (0) of DRV11/toneburst gate
CHNO	Integer array containing port (channel) numbers for each channel
ISW	Integer variable for returned status

Internal Summary:

<u>(LABEL)</u>	<u>(FUNCTION)</u>	<u>(PAGE)</u>
<u>SETUP ROUTINES</u>		
A32D	Argument transfer; parameter checking	4
SETDMA	Load DMA bus address and word count registers; load pseudo-register for DMA control	4
SETA2D	Load pseudo-register for A/D control; load DMA pseudo-registers into control status registers	6
<u>MEASUREMENT ROUTINE</u>		
READY	Prepare for measurement	7
GO	Check for pulsed-system gate	7
GOCW	Load A/D pseudo registers into actual control status registers; wait for completion (interrupt)	7

ADISR	Dummy interrupt service routine	8
DMISR	Functional interrupt service routine	8

```

1  .TITLE A32D
2  .IDENT /07/
3
4  SUBROUTINE TO LOAD AN INTEGER DATA ARRAY WITH DATA FROM UP
5  TO THREE ADAC-1012 A/D CONVERTERS, RUNNING SIMULTANEOUSLY.
6
7  AUTHOR: RICHARD E. SCOTT, JR. (DECEMBER 1981)
8
9  FORTRAN CALL--
10 CALL A32D (BUFFER,GAINS,NWORDS,ADS,CATED,CHNO,ISW)
11
12 "BUFFER" IS AN INTEGER ARRAY TO CONTAIN CONTIGUOUS DATA FROM
13 UP TO THREE CHANNELS OF DMA'ED DATA FROM THE A/D CONVERTERS.
14 EACH DATA WORD IS A 12-BIT INTEGER. FOR N WORDS PER CHANNEL...
15 CHANNEL 1 DATA IN BUFFER(1) TO BUFFER(N)
16 CHANNEL 2 DATA IN BUFFER(N+1) TO BUFFER(2N)
17 CHANNEL 3 DATA IN BUFFER(2N+1) TO BUFFER(3N)
18
19 "GAINS" ARE PROGRAMMABLE GAIN CODES (1,2,4,8) OR (1,2,5,10)
20 FOR A/D CONVERTERS (AS DEFINED BY HARDWARE STRAPPING).
21 GAINS IS AN INTEGER ARRAY.
22
23 "NWORDS" IS THE NUMBER OF WORDS PER CHANNEL TO BE DMA'ED
24 INTO MEMORY FROM THE A/D CONVERTERS.
25
26 "ADS" IS THE A/D SETUP ARRAY WHERE:
27 ADS(1) = A/D'S USED (#1=1) + (#2=2) + (#3=4) (ALL=7)
28 ADS(2) = DMA TO GET INTERRUPT FROM (1,2,3)
29 AGAIN, WITH ADS(1) = 3 ONE CAN USE TWO A/D CARDS ONLY WITHOUT
30 CONCERN FOR THE SUBROUTINE ADDRESSING THE NONEXISTENT THIRD
31 A/D CARDS.
32
33 "CATED" FLAGS THE PRESENCE OF PULSE/CATING MECHANISMS.
34 CATED = 1 MEANS USE PARALLEL I/O CARD TO FLAG START OF PULSE.
35 CATED = 0 MEANS DON'T.
36
37 THE LATTER CASE MEANS THAT A PARTICULAR BACKPLANE DOES NOT
38 NEED TO HAVE A DRV-11 INSTALLED TO AVOID TRAPS DUE TO ADDRESSING
39 UNNEEDED AND NONEXISTENT CARDS. HOWEVER, ONE MAY FIND OCCASIONAL
40 PHASE ERRORS BECAUSE OF CLOCK PULSES OCCURRING DURING THE SETUP
41 OF THE A/D REGISTERS. ONE TECHNIQUE USED IS TO REQUIRE THAT THE
42 HOST PROCESSOR TURN OFF THE SAMPLING CLOCK UNTIL THE REGISTERS
43 ARE ALL PREPARED.
44
45 "CHNO" IS AN ARRAY OF THREE CHANNEL NUMBERS, CHANNEL IN THE SENSE
46 OF THE SIXTEEN POSSIBLE MULTIPLEXED PORTS TO EACH A/D CONVERTER.
47
48 "ISW" IS STATUS OF TRANSFER (0 = AOK)
49 ISW = -1; A/D #1 ERROR (THESE 6 ERRORS ARE ADDITIVE)
50 ISW = -2; A/D #2 ERROR (DITTO...)
51 ISW = -4; A/D #3 ERROR
52 ISW = -10; DMA #1 ERROR
53 ISW = -20; DMA #2 ERROR
54 ISW = -40; DMA #3 ERROR (...RANGING -1 TO -77)
55 ISW = -100; PARAMETER ERROR (FROM THIS PROGRAM: A/D SETUP ERROR)
56 ISW = -101; PARAMETER ERROR (FROM CALLING PROGRAM: TOO MANY POINTS)
57
58 **** ALL ARGUMENTS ARE INTEGERS ****

```

```

59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108

;HARDWARE REGISTER DEFINITIONS

;REGISTERS FOR THE FIRST A/D CONVERTER (A):
ADCSRA=164040 ;A/D CONTROL STATUS REGISTER
ADDRBA=164042 ;A/D DATA BUFFER
DHWCR=172410 ;DMA WORD COUNT REGISTER
DHCARA=172412 ;DMA BUS ADDRESS REGISTER
DHCSRA=172414 ;DMA STATUS REGISTER
DNHUXA=172416 ;DMA CHANNEL DATA

;REGISTERS FOR THE SECOND A/D CONVERTER (B):
ADCSRB=164050 ;A/D CONTROL STATUS REGISTER
ADDRBB=164052 ;A/D DATA BUFFER
DNWCRR=172420 ;DMA WORD COUNT REGISTER
DNCARB=172422 ;DMA BUS ADDRESS REGISTER
DNCSRB=172424 ;DMA STATUS REGISTER
DNHUXB=172426 ;DMA CHANNEL DATA

;REGISTERS FOR THE THIRD A/D CONVERTER (C):
ADCSRC=164060 ;A/D CONTROL STATUS REGISTER
ADDRBC=164062 ;A/D DATA BUFFER
DHWCR=172430 ;DMA WORD COUNT REGISTER
DHCARC=172432 ;DMA BUS ADDRESS REGISTER
DNCSRC=172434 ;DMA STATUS REGISTER
DNHUXC=172436 ;DMA CHANNEL DATA

;A/D CONTROL STATUS REGISTER (ADCSR*):
;BIT 15: ERROR BIT (SET FOR CONVERSION TIMING ERROR)
;BIT 14: INTERRUPT ON ERROR (0)
;BIT 12-3: NOT USED (EXTENDED CHANNEL ADDRESS)
;BIT 11-8: ADDRESS OF CHANNEL (*DD)
;BIT 7: A/D DONE FLAG (SET AT END OF CONVERSION)
;BIT 6: INTERRUPT ENABLE (0)
;BIT 5: OPTIONAL ENABLE (0)
;BIT 4-3: PROGRAMMABLE GAIN (AS INPUT)
;BIT 2: SEQUENTIAL ENABLE
;BIT 1: EXTERNAL ENABLE (1)
;BIT 0: A/D CONVERSION GO BIT (0)

;DATA REGISTER (ADDR*):
;BITS 12-15: NOT USED (1111)
;BITS 00-11: DATA

;DMA WORD COUNT REGISTER (DNWCR*)
;BITS 12-15: NOT USED (1111)
;BITS 00-11: ONE'S COMPLEMENT OF WORDS

;DMA BUS ADDRESS (DNCAR*):
;BITS 00-11: WORD ADDRESS

```

```

110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140

;DMA CONTROL STATUS REGISTER (DMCSR*):
;BIT 15: ERROR BIT (SET FOR ANY ERROR)
;BIT 14: NONEXISTENT MEMORY ERROR
;BIT 13: ATTENTION BIT (EXTERNAL)
;BIT 12-8: NOT USED (0)
;BIT 7: READY FLAG (SET WHEN DMA DONE)
;BIT 6: INTERRUPT ENABLE (0)
;BIT 5-4: EXTENDED ADDRESS BITS
;BIT 3-1: NOT USED (0)
;BIT 0: DMA CO BIT (1)

;REGISTERS FOR THE DRV-11 USED FOR SYNCHRONIZATION IN PULSED
;SYSTEMS.
;
;DRCSR=167770
;DROUT=167772
;DRINN=167774
;
;VECTOR ADDRESSES:
; 130 - A/D COMPLETION (A) OR ERROR
; 150 - DMA COMPLETION (A)
; 170 - A/D COMPLETION (B) OR ERROR
; 210 - DMA COMPLETION (B)
; 230 - A/D COMPLETION (C) OR ERROR
; 250 - DMA COMPLETION (C)
;
;SET PROM = 1 IF VECTOR SPACE IS WITHIN ROM SPACE AND SET
;VECTORS REQUIRED. OTHERWISE, VECTORS ARE LOADED DYNAMICALLY.
;SET PROM = -1 FOR RAM VARIABLES ASSEMBLED LOCALLY.
PROM=1
000001

```

```

142 ;SUBROUTINE INITIALIZATION:
143 ; GRAB THE ARGUMENTS.
144 ; DO SOME CHECKING FOR VALIDITY.
145
146 A32D::
147
148 TST @12(R5)
149 BEQ 10$
150 CLR @DRCSR
151
152 ;START OF ROUTINE
153 ;IF HAVE, THEN...
154 ;CLEAR UP THE DRV-11 STATUS REGISTER
155 ;ADDRESS OF ADS
156 ;A/D'S-USED WORD
157 ;DMA INTERRUPT WORD
158 ;ADDRESS OF FIRST DATA BUFFER
159 ;NUMBER OF WORDS
160 ;BUT ONE'S COMPLEMENTED
161 ;ADDRESS OF CHANNEL PORTS
162
163 ;CHECK FOR ERRORS
164 ;WAD > 7
165 ;WAD < 1
166 ;IWD > 3
167 ;IWD >= 1
168
169 ;PARAMETER ERROR
170
171 ;SET UP THE NECESSARY REGISTERS FOR THOSE A/D-DMAS
172 ;THAT ARE TO BE USED.
173
174 ;FIRST DMA REGISTERS SETUP:
175
176 BIT #1,WAD
177 BEQ 20$
178 R4,@DMACRA
179 R0,@DMCARA
180 #1,@T1
181 (R2)+,R1
182 R1,@DMMUXA
183 SWAB R1
184 MOV R1,@ADD
185 .IF EQ PROM
186 MOV #ADISR,@130
187 MOV #340,@132
188 MOV #DNISR,@150
189 MOV #340,@152
190 .ENDC
191
192 CHIP #1,IWD
193 BNE 20$
194 MOV #101,@T1
195 .IF EQ PROM
196 MOV #DNISR,@150
197 .ENDC

```

```

199
200
201 000176 067500 000006
202 000202 067500 000006
203 000206 032767 000002 000000G
204 000214 001424
205 000216 010437 172420
206 000222 010037 172422
207 000226 012737 000001 000000G
208 000234 012201
209 000236 010137 172426
210 000242 000301
211 000244 010137 000000G
212
213
214
215
216
217
218 000250 022767 000002 000000G
219 000256 001003
220 000260 012737 000101 000000G
221
222
223
224
225
226
227 000266 067500 000006
228 000272 067500 000006
229 000276 032767 000004 000000G
230 000304 001426
231 000306 010437 172430
232 000312 010037 172432
233 000316 012737 000001 000000G
234 000324 012201 172436
235 000326 010137
236 000332 000301 170377
237 000334 042701 000000G
238 000340 010137
239
240
241
242
243
244
245 000344 022767 000003 000000G
246 000352 001003
247 000354 012737 000101 000000G
248
249
250

```

!SECOND DMA REGISTERS SETUP

```

ADD @6(R5),R0 !INCREASE BUFFER ADDRESS
ADD @6(R5),R0 !(TWICE FOR BYTES)
BIT #2,WAD !IS THIS DMA ACTIVE?
BEQ 30$ !NOPE!
MOV R4,@#DMWCRB !LOAD WORD COUNT REGISTER
MOV R0,@#DMCARB !LOAD ADDRESS REGISTER
MOV #1,@#T2 !LOAD CSR, GO ONLY
MOV (R2)+,R1 !GET PORT NUMBER
MOV R1,@#DMMUXB !LOAD DMA MULTIPLEXOR ADDRESS
SVAB R1,@#BDD !SHIFT INTO BITS 8-11
!THEN STORE FOR LATER USE
!IF EQ PROM
MOV #ADISR,@#170 !LOAD INTERRUPT VECTOR (A/D)
MOV #340,@#172 !AND PROCESSOR STATUS WORD
MOV #DMISR,@#210 !LOAD INTERRUPT VECTOR
MOV #340,@#212 !AND PROCESSOR STATUS WORD
.ENDC
CHP #2,IWD !INTERRUPT OFF THIS ONE?
BNE 30$ !NOPE!
MOV #101,@#T2 !LOAD CSR, IE AND GO
!IF EQ PROM
MOV #DMISR,@#210 !LOAD ACTIVE ISR VECTOR
.ENDC

```

!THIRD DMA REGISTERS SETUP

```

ADD @6(R5),R0 !INCREASE BUFFER ADDRESS
ADD @6(R5),R0 !(TWICE FOR BYTES)
BIT #4,WAD !IS THIS DMA ACTIVE?
BEQ SETA2D !NOPE!
MOV R4,@#DMWCRB !LOAD WORD COUNT REGISTER
MOV R0,@#DMCARB !LOAD ADDRESS REGISTER
MOV #1,@#T3 !LOAD CSR, GO ONLY
MOV (R2)+,R1 !GET PORT NUMBER
MOV R1,@#DMMUXC !LOAD DMA MULTIPLEXOR ADDRESS
SVAB R1 !SHIFT INTO BITS 8-11
BIC #170377,R1 !CLEAN UP ACCIDENTS
MOV R1,@#CDD !THEN STORE FOR LATER USE
!IF EQ PROM
MOV #ADISR,@#230 !LOAD INTERRUPT VECTOR (A/D)
MOV #340,@#232 !AND PROCESSOR STATUS WORD
MOV #DMISR,@#250 !LOAD INTERRUPT VECTOR
MOV #340,@#252 !AND PROCESSOR STATUS WORD
.ENDC
CHP #3,IWD !INTERRUPT OFF THIS ONE?
BNE SETA2D !NOPE!
MOV #101,@#T3 !LOAD CSR, IE AND GO
!IF EQ PROM
MOV #DMISR,@#250 !LOAD ACTIVE ISR VECTOR
.ENDC

```

```

252      !PREPARE AND LOAD THE A/D CSR'S WITH GAIN, ETC.--EVERYTHING
253      !EXCEPT THE GO BITS. ALSO LOAD THE DMA CSR'S PREVIOUSLY CHOSEN.
254      !LOAD R0-R3 WITH THE ADDRESSES OF THE ACTIVE A/D CSR'S (OR A
255      !DUMMY ADDRESS IF NOT). THIS WILL ALLOW QUICK SETUP-AND-GO
256      !OF THE A/D'S (NEAR SIMULTANEOUS) WITHOUT THE LOGIC REQUIRED
257      !TO NOT "DO" A/D CSR'S NOT EXISTING.
258
259      MOV      4(R5),R3      !ADDRESS OF GAINS ARRAY
260      MOV      610(R5),WAD    !GET A/D APPLICABILITY WORD
261
262      !SET UP A/D CONVERTER NUMBER ONE IF ACTIVE
263
264      MOV      *ADUM,R0      !TARGET ADDRESS IF NOT ACTIVE
265      BIT      *1,WAD        !IS THIS A/D ACTIVE?
266      BEQ      30$          !NOPE
267      MOV      (R3)+,R0      !GET GAIN CODE OF CHANNEL A (1,2,5,10)
268      ASR      R0           !CHANGE TO 0,1,2,5
269      CHP      *5,R0        !CHECK IF 5
270      BCT      20$          !NOPE
271      DEC      R0           !SET 5 TO 4
272      DEC      R0           !THEN TO 3
273      ASH      *3,R0        !SHIFTED INTO BITS 3-4
274      CON      R0           !SET TO 30,20,10,00
275      BIC      *177747,R0   !SAFETY CLEANUP
276      BIS      @*ADD,R0     !ADD ADDRESS SETTING TO R0
277      MOV      R0,@*ADCSRA  !SET GAIN (ETC) BITS FIRST
278      MOV      *ADCSRA,R0   !TARGET ADDRESS IF ACTIVE
279      MOV      T1,@*DMCSRA  !ENABLE DMA'S
280
281      !SET UP A/D CONVERTER NUMBER TWO IF ACTIVE
282
283      MOV      *ADUM,R1      !TARGET ADDRESS IF NOT ACTIVE
284      BIT      *2,WAD        !IS THIS A/D ACTIVE?
285      BEQ      50$          !NOPE
286      MOV      (R3)+,R1      !GET GAIN CODE OF CHANNEL B (1,2,5,10)
287      ASR      R1           !CHANGE TO 0,1,2,5
288      CHP      *5,R1        !CHECK IF 5
289      BCT      40$          !NOPE
290      DEC      R1           !SET 5 TO 3
291      DEC      R1           !SHIFTED INTO BITS 3-4
292      ASH      *3,R1        !SET TO 30,20,10,00
293      CON      R1           !ADD ADDRESS SETTING TO R1
294      BIC      *177747,R1   !SET GAIN (ETC) BITS FIRST
295      BIS      @*BDD,R1     !TARGET ADDRESS IF ACTIVE
296      MOV      R1,@*ADCSRB  !ENABLE DMA'S
297      MOV      *ADCSRB,R1
298      MOV      T2,@*DMCSRB

```

```

300
301 000550 012702 000000C 508:
302 000554 032767 000004 000000C
303 000562 001425
304 000562 012302
305 000564 006202 000005
306 000566 022702
307 000570 003002
308 000574 005302
309 000576 005302
310 000600 005302
311 000602 072227 000003 608:
312 000606 005102
313 000610 042202 177747
314 000614 053702 000000C
315 000620 010237 000000C
316 000624 012702 164060
317 000630 016737 000000C 172434
318
319
320
321
322
323
324 000636 005067 000000C
325 000642 005075 000016
326
327 000646 000240
328 000650 005775 000012
329 000654 001404
330 000656 032737 000001 167774
331 000664 001774
332
333
334
335
336
337
338
339
340
341
342 000666 012703 000003
343 000672 050310
344 000674 050311
345 000676 050312
346 000676 050312
347
348 000700 000001
349 000702 005767 000000C
350 000706 000240
351 000710 001773
352
353 000712 000207

```

;SET UP A/D CONVERTER NUMBER THREE IF ACTIVE
 ;TARGET ADDRESS IF NOT ACTIVE
 ;IS THIS A/D ACTIVE?
 ;NOPE
 ;GET GAIN CODE OF CHANNEL C (1,2,5,10)
 ;CHANGE TO 0,1,2,5
 ;CHECK IF 5
 ;NOPE
 ;SET 5 TO 3
 ;SHIFTED INTO BITS 3-4
 ;SET TO 30,20,10,00
 ;ADD ADDRESS SETTING TO R2
 ;SET GAIN (ETC) BITS FIRST
 ;TARGET ADDRESS IF ACTIVE
 ;ENABLE DMA'S

;IS EVERYBODY READY??
 ;INCLUDING TRANSMIT GATE TO DLV-11???
 CLR ADST
 CLR 016(R5)
 NOP
 TST 012(R5)
 BEQ COCW
 BIT 01,0-DRINN
 BEQ CO
 ;USE DRV-11 FOR PULSE GATE?
 ;NO
 ;CHECK FOR BIT 0 SET
 ;...AND WAIT...

;THE FOLLOWING IS DONE THIS WAY SO THE LOADING IS AS FAST
 ;AS POSSIBLE, THUS ELIMINATING ANY POSSIBLE PHASE ERRORS
 ;BETWEEN THE THREE CHANNELS.
 ;TIME TO LOAD EACH CSR IS ABOUT 4.54 MICROSECONDS.
 ;NOTE THAT IF ACTIVE, THE DESTINATION REGISTERS CONTAIN THE
 ;ADDRESSES OF THE A/D CSR'S; IF NOT ACTIVE, THEY CONTAIN A
 ;DUMMY ADDRESS TO AVOID THE POSSIBILITY OF WRITING TO A
 ;NON-EXISTENT ADDRESS (AND TRAPPING) IF INACTIVE A/D CONVERTERS
 ;ARE NOT IN THE BACKPLANE.

;START A/D CONVERSIONS
 ;WAIT FOR SPECIFIED INTERRUPT
 ;...AND WAIT...
 ;RETURN WITH DATA

- (3) MODULE CONVRT - Subroutine to convert a floating-point number to an ASCII string.

Calling Sequence:

CALL CONVRT (VALUE, STRING, K, L)

where the arguments are:

VALUE	Floating-point variable to be encoded
STRING	Byte array to load encoded variable
K	Encode with K places to left of decimal point
L	Encode with L places to right of decimal point

NOTE: Decimal point (.) and sign (+ or -) are not included in the size of K and L.

```

1      .TITLE CONVRT
2      .IDENT /01/
3
4      ;SUBROUTINE TO CONVERT A REAL NUMBER (FLOATING) TO
5      ;AN ASCII STRING OF FORMAT "KK.LLL" FOR PRINTING
6      ;PURPOSES.
7
8      ;CALL CONVRT (VALUE,STRING,K,L)
9      ;WHERE VALUE BECOMES AS F<K>.<L> (FORTRAN)
10
11      ;AUTHOR: RICHARD E. SCOTT, JR. (NHL/USRD -- AUGUST 1981)
12
13      AC0=%0
14      AC1=%1
15
16      CONVRT:
17          MOV     R0,-(SP)
18          MOV     R1,-(SP)
19          MOV     R2,-(SP)
20          MOV     R3,-(SP)
21          MOV     R4,-(SP)
22
23          SETF    4(R5),R3
24          MOV     LDF    62(R5),AC0
25          CFCC    BGT     10$
26          MOV     R'--, (R3)+
27          BR      20$
28          MOV     R'++, (R3)+
29          ABSF    AC0
30          STCFI   AC0,R1
31          LDCIF   R1,AC1
32          MOV     66(R5),R4
33          JSR     PC,DECO
34          MOV     R'--, (R3)+
35          SUBF    AC1,AC0
36          MOV     610(R5),R4
37          MOV     67TEN,AC0
38          DEC     R4
39          BGT     30$
40          ADDF    67R0,AC0
41          STCFI   AC0,R1
42          MOV     610(R5),R4
43          JSR     PC,DECO
44
45          MOV     (SP)+,R4
46          MOV     (SP)+,R3
47          MOV     (SP)+,R2
48          MOV     (SP)+,R1
49          MOV     (SP)+,R0
50
51          RTS
52
53          PC

```

CONVRT MACRO M1113 23-JUL-82 14:14 PAGE 2

55	000134	010402		MOV	R4,R2	DATA IN R1
56	000134	060203		ADD	R2,R3	CHARACTER COUNTER
57	000136	006700		SXT	R0	POINTER--WORK RIGHT TO LEFT
58	000140	006700	000012	DIV	10,R0	EXTEND VALUE TO 32 BITS
59	000142	062701	000060	ADD	60,R1	DIVIDE BY 10
60	000146	110143		MOVB	R1,--(R3)	CONVERT REMAINDER TO ASCII
61	000152	010001		MOV	R0,R1	AND STORE IN STRING
62	000154	003367		DEC	R2	CONVERT QUOTIENT TO DIVIDEND
63	000156	005302		BGT	10	DONE?
64	000160	003367		ADD	R4,R3	NO
65	000162	060403		RTS	PC	RESET STRING POINTER
66	000164	000207				
67						
68	000166	041040	000000	.FLT2	10.	
69	000172	037777	171345	.FLT2	0.4999	
70						
71		000001		.END		

- (4) MODULE DAPD - Subroutine to compute voltage and phase from the real and imaginary parts of the voltage, and to compute the harmonic distortion from a specific number of harmonics.

Calling Sequence:

CALL DAPD (X, N, K, AMP, PHASE, DSTORT, LHARM, IERR)

where the arguments are:

X	Floating-point data sequence
N	Length of data sequence
K	Spectral line (number of cycles) to compute
AMP	Resulting total rms voltage
PHASE	Resulting phase in degrees
DSTORT	Harmonic distortion
LHARM	Number of harmonics used to compute distortion
IERR	Execution status

- NOTES:
1. If LHARM = 1, distortion is not computed
 2. If DSTORT = -1, distortion was not computed
 3. If DSTORT = -2, error occurred (AMP = 0)
 4. If IERR = -1, K = N (illegal condition)
 5. If IERR = LHARM, harmonic folds onto dc line

```

1  .TITLE DAPD
2  .IDENT /01/
3  ;AHPHIA.17N 17-JUL-81 JDC GEORGE (ORIGINATOR)
4  ;DAPD .MAC 02-OCT-81 RE SCOTT (TRANSLATOR)
5
6  ;SUBROUTINE DAPD (X,N,K,AMP,PHASE,DSTORT,LHARM,IERR)
7
8  ;COMPUTE AMPLITUDE, PHASE, & PERCENT HARMONIC DISTORTION
9  ;FOR THE K-TH SPECTRAL LINE OF THE DFT OF X, A REAL SEQUENCE
10 ;OF LENGTH N.
11 ;UNDERSAMPLING OR FOLDING WHERE K CAN EXCEED N/2 IS ALLOWED.
12 ;SUBROUTINES REQUIRED: DGZL (X,N,K,XR,XI)
13
14 ;REAL X(N)
15 ;INTEGER N
16 ;INTEGER K
17 ;DATA SEQUENCE
18 ;DFT SPECTRAL LINE NUMBER OF INTEREST
19 ;ALSO # CYCLES OF TEST SINUSOID IN N SAMPLES
20 ;I.E. K CYCLES OF FUNDAMENTAL (1ST HARM) IN SEQUENCE;
21 ;2K CYCLES OF 2ND HARMONIC, ETC.
22 ;IF K=0, COMPUTE DC COMPONENT.
23 ;UNS MAGNITUDE OF K TH SPECTRAL COMPONENT
24 ;PHASE IN DEGREES OF K-TH SPECTRAL COMPONENT
25 ;MODEL IS 0 DEG COSINE -90 DEG SINE
26 ;% HARMONIC DISTORTION OF K TH COMPONENT
27 ;FOR LINE #'S 2*K THROUGH LHARM*K
28 ;HARM=0 SKIP DISTORTION CALCULATION; SAVE TIME
29 ;IERR=0 NORMAL
30 ;IERR=-1 IF (MOD(K,H)=0)
31 ;IERR=-L IF HARMONIC FOLDS ON DC LINE
32
33 ;DSTORT = -1 IF NOT COMPUTED
34 ;DSTORT = -2 IF ERROR IN COMPUTING
35
36 AC0=%0
37 AC1=%1
38 AC2=%2
39
40 HARMIN = 11. ;MAXIMUM HARMONICS TO USE
41
42 ;EXPERIENCE SEEMS TO SHOW THAT A MINIMUM OF 9 POINTS ARE
43 ;REQUIRED TO COMPUTE ACCURATELY THE AMPLITUDE AND PHASE OF
44 ;A SINUSOID; FOR DISTORTION, 2*LHARM+1 POINTS ARE REQUIRED.
45
46 ;SET PROH = 1 FOR RAM VARIABLES STORED EXTERNALLY
47 ;SET PROH = 0 FOR RAM VARIABLES STORED INTERNALLY (NORMAL)
48 PROH=1
49
50 DAPD::
51
52 NOV R0,-(SP)
53 NOV R1,-(SP)
54 NOV R2,-(SP)
55 NOV R3,-(SP)
56 NOV R4,-(SP)
57
58 SETP

```



```

107
108 000144 170001
109 000146 172475 000010
110 000152 174002
111 000154 171000
112 000156 172575 000012
113 000162 171101
114 000164 172001
115 000166 172000
116 000170 054752 0000000G
117 000174 012605
118 000176 174975 000010
119
120 000202 170502
121 000204 170000
122 000206 091420
123
124 000210 010546
125 000212 012705 000720'
126 000216 004767 0000000G
127 000222 172400
128 000224 171037 000670'
129 000230 005767 0000000G
130 000234 003001
131 000236 170700
132 000240 012605
133 000242 174075 000012
134 000246 000404
135
136 000250 172637 000700'
137 000254 174275 000012
138

; *; AMP = SQRT(2*(XR*XR + XI*XI))
; COMPUTE VOLTAGE AND PHASE
; XR
; SAVE FOR TEST
; XR*XR
; XI
; XI*XI
; XR*XR + XI*XI
; 2 (XR*XR + XI*XI)
; = VDF
; RESTORE R5

; WATCH FOR ATAN(XI/0)
; ERROR!
; *; IF (XR.NE.0)
; *; PHASE=PSIGN*SATAN2(XI,XR)*180./3.1415927
; SAVE R5 (FOR CALL SATAN2)

; TAN-1 (XI/XR)
; CONVERT TO DECREES
; WAGON WHEELS?
; FORWARD
; NO--BACKWARD ROTATING
; RESTORE R5
; PDEF

; *; IF (XR.EQ.0) PHASE=0.0
; FOR ATAN(XI/0)...
; SET PDEF = 90

```



```

153
159 060412 013701 000000G
191 060416 017502 000000+
192 060422 006202
195 060424 020102
196 060426 063003
195 060430 010137 000000G
195 060434 000405
197 060436 167501 000000+
198 060442 005401 208:
199 060444 010137 000000G
200
201 060450 012702 308:
202
203 060454 012700 353:
204 060460 050300
205 060462 060200
206 060464 021001
207 060466 001437
208
209 060470 005302
210 060472 020203
211 060474 002707
212
213 060476 012700 177776G
214 060502 060306
215 060504 060300
216 060506 010110
217
218
219 060510 010546
220 060512 010446
221 060514 010346
222
223
224
225 060516 012705 000654+
226 060522 004707 000000G
227
228
229 060526 172475 000010
230 060532 171000
231 060534 172575 000012
232 060540 171101
233 060542 172001
234 060544 172000
235 060546 172537 000000G
236 060552 172100
237 060554 174137 000000G
238
239 060560 012603
240 060562 012604
241 060564 012605

MOV R5,R1
MOV R4,R5,R2
ASR R2
CHP R1,R2
RCT 208
MOV R1,R1
IR 308
SUB R4,R5,R1
NEG R1
MOV R1,R1
MOV R1,R2
MOV R2,R0
ADD R2,R0
ADD R2,R0
CHP (R0),R1
BEQ 408
INC R2
CHP R2,R3
BLT 353
MOV R3,R0
ADD R3,R0
ADD R3,R0
MOV R1,(R0)

MOV R5, -(SP)
MOV R4, -(SP)
MOV R3, -(SP)
IF EQ PROH
MOV R2,R5, 0,DCARG+2
ENDC
MOV R5,DCARG,R5
JSR PC,BUZZL

LDF R10(R5),AC0
MULF AC0,AC0
LDF R12(R5),AC1
MULF AC1,AC1
ADDF AC1,AC0
ADDF AC0,AC0
LDF R2PVR,AC1
ADDF AC0,AC1
STF AC1, 0,PVR
(PVR)+,R3
(PVR)+,R4
(PVR)+,R5

;IF (1,LE,N/2) I=1
;IF (1,GT,N/2) I=N-1
;RESTORE I TO R1
;GET NSAM
;NSAM / 2
;COMPARE I AND NSAM/2
;I > NSAM/2
;I < NSAM/2 SO KEEP I
;GET I - NSAM
;NSAM - I
;USE I = NSAM-I
;LL=1
;DO 20 LL = 1,LL
;IF (HARM(LL),EQ,I) GOTO 30
;HARM(-1) ADDRESS
;HARM(LL) ADDRESS
;IS IT I(=R1)?
;IS USED
;CONTINUE
;LL=LL+1
;R3 IS STILL "L"
;L < LL
;HARM(L) = I
;HARM(-1) ADDRESS
;HARM(L) ADDRESS
;I = 1

;CALL DCITZL (X,N,I,XR,XI)
;SAVE R5
;SAVE R4
;SAVE R3
;TRANSFER ADDRESS OF ARRAY
;LOAD ARGUMENT LIST
;DFT THE HARMONIC
;PVR=PVR+2*(XI*XR+XI*XI)
;XR
;XI*XR
;XI
;XI*XI
;XR*XR + XI*XI
;2 (XR*XR + XI*XI)
;SUB TO PVR
;AND SAVE
;RESTORE R3
;RESTORE R4
;RESTORE R5

```

```

243 000566 005304
244 000570 03263
245 000572 172475 000010
246 000576 170900
247 000580 003413
248 000584 171000
249 000588 174560
250 000592 172475
251 000596 172475
252 000600 172475
253 000604 172475
254 000608 172475
255 000612 172475
256 000616 172475
257 000620 172475
258 000624 172475
259 000628 172475
260 000632 172475
261 000636 172475
262 000640 172475
263 000644 172475
264 000648 172475
265 000652 172475
266 000656 172475
267 000660 172475
268 000664 172475
269 000668 172475
270 000672 172475
271 000676 172475
272 000680 172475
273 000684 172475
274 000688 172475
275 000692 172475
276 000696 172475
277 000700 172475
278 000704 172475
279 000708 172475
280 000712 172475
281 000716 172475
282 000720 172475
283 000724 172475
284 000728 172475
285 000732 172475
286 000736 172475
287 000740 172475
288 000744 172475
289 000748 172475
290 000752 172475
291 000756 172475
292 000760 172475
293 000764 172475

```

400: DEC R4
 401: BDT 105
 402: LDF @10(R5),AC0
 403: CFCC B03
 404: RULF AC0,AC0
 405: DIVF AC0,AC1
 406: LDF AC1,AC0
 407: JSR PC,8*SQRT
 408: LDF @*CENT,AC1
 409: RULF AC1,AC0
 410: STF AC0,@14(R5)
 411: DR DONE
 412: LDF @-R2,AC0
 413: STF AC0,@14(R5)
 414: DONE
 415: H0V (SP)+,R4
 416: H0V (SP)+,R3
 417: H0V (SP)+,R2
 418: H0V (SP)+,R1
 419: H0V (SP)+,R0
 420: RTS PC
 421: 5, RBUF, NSAN, 1, XREAL, XIMAG
 422: .FLT2 57.2950 ; RADIANS-TO-DEGREES CONVERSION FACTOR
 423: .FLT2 100.
 424: .FLT2 90.
 425: .FLT2 -1.
 426: .FLT2 -2.
 427: .FLT2 -3.
 428: .WORD 2
 429: .WORD XIMAG
 430: .WORD XREAL
 431: .IF EQ FROM
 432: ; RAM VARIABLES USED BY DAPD
 433: .FLT2 0.0
 434: .FLT2 0.0
 435: .FLT2 0.0
 436: .WORD 0
 437: .WORD 0
 438: .WORD 0
 439: .WORD 0
 440: .WORD 0
 441: .BLKW HARLIN ; ARRAY FILLED AS HARMONICS ARE USED
 442: .WORD 0
 443: .ENDC
 444: .END
 000001

- (5) MODULE DGZL - Subroutine to perform a discrete Fourier transform on a data sequence using Goertzel's algorithm.

Calling Sequence:

CALL DGZL (X, N, K, XR, XI)

where the arguments are:

X	Floating-point data sequence
N	Length of sequence
K	Spectral line (number of cycles) to compute
XR	Real part of voltage
XI	Imaginary part of voltage

```

1 .TITLE DCZL
2 .IDENT /01/
3
4 ;SUBROUTINE DCZL (X,N,K,XR,XI)
5
6 ;REAL X(N)
7 ;SEQUENCE LENGTH
8 ;INTEGER N
9 ;INTEGER K
10 ;ALSO # CYCLES OF TEST SINUSOID IN N SAMPLES
11 ;REAL XR
12 ;REAL XI
13 ;IMAGINARY PART OF VOLTAGE
14
15 ;DCRTZL.FTN -- JD GEORGE (17-JUL-81) (ORIGINATOR)
16 ;DCZL .MAC -- RE SCOTT (02-OCT-81) (TRANSLATOR)
17
18 ;COERTZEL'S ALGORITHM
19 ;TO COMPUTE K-TH LINE OF N POINT REAL SEQUENCE X(L)
20 ;REFERENCE: G. COERTZEL
21 ;"AN ALGORITHM FOR THE EVALUATION OF FINITE TRIG SERIES"
22 ;AMERICAN MATHEMATICAL MONTHLY, V65, 1958, PP.34-35
23 ;ALSO CHAPTER 24, PP.258-62
24 ;MATHEMATICAL METHODS FOR DIGITAL COMPUTERS VOL. 1
25 ;EDITED BY A. RALSTON & H.S. WILF
26 ;JOHN WILEY & SON, 1967
27
28 ;A DOUBLE PRECISION COERTZEL ROUTINE WHICH IS MORE ACCURATE AND
29 ;FASTER THAN "SPDFT.FTN"
30 ;40% RUNNING TIME OF SPDFT.FTN
31 ;UP TO 512 POINTS AMPLITUDE ERROR < 0.0012%
32 ;PHASE ERROR < 0.00001DEG
33
34 AC0=%0
35 AC1=%1
36 ULP1=%2
37 ULP2=%3
38
39 ;FLOATING POINT ACCUMULATOR #0
40 ;FLOATING POINT ACCUMULATOR #1
41 ;FLOATING POINT ACCUMULATOR #2
42 ;FLOATING POINT ACCUMULATOR #3
43
44 ;SET PROM = 1 FOR RAM VARIABLES STORED EXTERNALLY
45 ;SET PROM = 0 FOR RAM VARIABLES STORED INTERNALLY. (NORMAL)
46 PROM=1
47
48 DCZL::
49 MOV R5,-(SP)
50 SETD SETI
51
52 LDC1D @6(R5),AC0
53 LDC1D @4(R5),AC1
54
55 DIVD AC1,AC0
56 MULD @PI,AC0
57 ADDD AC0,AC0
58 STD AC0,@PI
59
60 JSR PC,@SDCOS
61 STD AC0,@PI
62 ADDD AC0,AC0
63 STD AC0,@PI
64
65 ;C0=DCOS(A0)
66 ;C1=2.*C0
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

58	000050	172437	000000C	LDD	@A0,AC0	!S0=DSIN(A0)
59	000054	004767	000000C	JSR	PC,00DSIN	
60	000060	174037	000000C	STD	AC0,e/S0	
61						
62	000064	170402		CLRD	ULP1	!ULP1=0.0D0
63	000066	170403		CLRD	ULP2	!ULP2=0.0D0
64						
65	000070	012605		MOV	(SP)+,R5	!RESTORE ARGUMENT LIST
66	000072	017501	0000004	MOV	@4(R5),R1	!COUNTER N IN R1
67	000076	010100		MOV	R1,R0	!DATA POINTER IN R0
68	000100	005301		DEC	R1	!COUNT FROM N TO 2
69	000102	006300		ASL	R0	
70	000104	006300		ASL	R0	!REAL*4 DATA BUFFER
71	000106	006500	0000002	ADD	2(R5),R0	!ADDRESS OF LAST+1 DATA WORD
72						
73						
74	000112	172437	000000C	LDD	@C1,AC0	!DO 10 L=N,2,-1
75	000116	171002		MULD	ULP1,AC0	!UL=X(L)+C1*ULP1-ULP2
76	000120	173003		SUBD	ULP2,AC0	
77	000122	177540		LDCFD	-(R0),AC1	
78	000124	172001		ADDD	AC1,AC0	
79	000126	172702		LDD	ULP1,ULP2	!ULP2=ULP1
80	000130	172600		LDD	AC0,ULP1	!ULP1=UL
81	000132	077111		SQB	R1,100	
82						
83	000134	172437	000000C	LDD	@C0,AC0	!XR=(X1+C0*ULP1-ULP2)/(N)
84	000140	171002		MULD	ULP1,AC0	
85	000142	173003		SUBD	ULP2,AC0	
86	000144	177575	0000002	LDCFD	@2(R5),AC1	
87	000150	172001		ADDD	AC1,AC0	
88	000152	177175	0000004	LDCID	@4(R5),AC1	
89	000156	174401		DIVD	AC1,AC0	
90	000160	176075	0000010	STCDF	AC0,@10(R5)	
91						
92	000164	172437	000000C	LDD	@S0,AC0	!XI=-S0*ULP1/(N)
93	000170	171002		MULD	ULP1,AC0	
94	000172	174401		DIVD	AC1,AC0	
95	000174	170700		NECD	AC0	
96	000176	176075	0000012	STCDF	AC0,@12(R5)	
97						
98	000202	000207		RTS	PC	
99	000204	040511	007732 121041	.FLT4	3.1415926535897932	
100	000212	064301				
101				.IF EQ	PROM	
102				.FLT4	0.0	
103				.FLT4	0.0	
104				.FLT4	0.0	
105				.FLT4	0.0	
106				.ENDC		
				.END		

(6) MODULE GPIOSR - This subroutine controls input and output along the IEEE-488 bus. There are three primary functions:

GPINI Initialize the interface and requisite variables; also set up address to use when group execute trigger (GET) is received.

Calling Sequence:

CALL GPINI (RESET)

where the argument is:

RESET Address of routine to transfer to on GET

GPIN Listen function - read a byte string from the IEEE-488 bus, terminated by a character received with EOI true, or by input buffer overflow.

Calling Sequence:

CALL GPIN (INBUF, MAXCH, NCH)

where the arguments are:

INBUF Byte buffer for input string

MAXCH Maximum number of characters allowed

NCH Number of characters received

GPOUT Talk function - set SRQ and transmit a byte string to the IEEE-488 bus.

Calling Sequence:

CALL GPOUT (OUTBUF, NCH)

where the arguments are:

OUTBUF Byte buffer for output string

NCH Number of characters to output

Also included is an interrupt service routine. The only interrupt always enabled is the group execute trigger (GET) interrupt, which feature is used to effect a program abort or restart function. The bus-in and bus-out interrupts are enabled only when their respective subroutines are called and disabled on completion of transfer.

The stand-alone driver supplied by National Instruments Corp. was not used for two reasons: size (keeping the total program size under 4K) and to implement the special abort function. However, programming this complex interface was not a straightforward exercise.

AD-A126 980

A MICROCOMPUTER-BASED SAMPLING DIGITAL VOLTMETER(U)
NAVAL RESEARCH LAB WASHINGTON DC R E SCOTT ET AL.
31 MAR 83 NRL-MR-4872

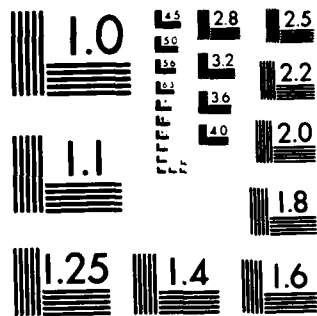
2/2

UNCLASSIFIED

F/G 14/2

NI





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```

1  TITLE CPIOSR
2  IDENT /08/
3
4  SUBROUTINES FOR HANDLING THE GPIBV-11 INTERFACE
5  GPIV1 -- INITIALIZE BUS
6  GPIV2 -- INPUT DATA FROM BUS (LISTEN)
7  GPIV3 -- OUTPUT DATA TO BUS (TALK)
8  GPIV4 -- INTERRUPT SERVICE ROUTINE
9
10  AUTHOR: R. E. SCOTT, JR. (DECEMBER 1981)--V05
11  MODIFIED TO USE EOI ON READ (FEBRUARY 1982)--V06
12  MODIFIED TO USE R5 ARGUMENT LIST (FEBRUARY 1982)--V07
13  MODIFIED TO USE TRIGGER (NOT "AB") FOR ABORT--V08
14  INRL/UNDERWATER SOUND REFERENCE DETACHMENT
15
16  INITIALIZE--CALL GPIV1 (RESET)
17  INITIALIZES THE GPIBV-11 INTERFACE.
18  LOADS THE INTERRUPT VECTOR AND PSW (IF ALLOWED).
19  LOADS THE "RESET" ADDRESS (GO TO IF "GET" RECEIVED),
20  AND ENABLES BOTH INPUT AND OUTPUT INTERRUPTS.
21
22  INPUT--CALL GPIV2 (INBUF,MAXCH,NCH)
23  ALLOWS A STRING OF VARIABLE LENGTH "NCH" (BUT LESS THAN
24  "MAXCH") TO BE LOADED IN "INBUF". EOI MUST COME WITH LAST BYTE.
25
26  OUTPUT--CALL GPIV3 (OUTBUF,NCH)
27  OUTPUTS A STRING "OUTBUF" OF VARIABLE LENGTH "NCH"
28  TO THE GPIB BUS.
29
30  INTERRUPT SERVICE ROUTINE--
31  CHECKS FOR INPUT OR OUTPUT INTERRUPTS AND HANDLES WHICHEVER
32  IS APPROPRIATE. IT ALSO CHECKS FOR ONE SPECIAL INPUT, THE
33  RESET (ABORT) FUNCTION, WHICH REINITIALIZES EVERYTHING.
34
35  GPIB REGISTERS REQUIRED:
36  GPVEC = 270 ; INTERRUPT VECTOR ADDRESS
37  GPISR = 164000 ; INTERRUPT SERVICE REGISTER (RO)
38  GPINR = 164000 ; INTERRUPT MASK REGISTER (WO)
39  GPCTSR = 164001 ; CONTROL STATUS REGISTER (RO)
40  GPCSR = 164002 ; COMMAND STATUS REGISTER (RO)
41  GPADR = 164004 ; ADDRESS MODE REGISTER (WO)
42  GPACR = 164006 ; AUXILIARY CONTROL REGISTER (RW)
43  GPASNR = 164010 ; ADDRESS SWITCH REGISTER (RO)
44  GPADR = 164010 ; ADDRESS REGISTER (WO)
45  GPSPR = 164012 ; SERIAL POLL REGISTER (RW)
46  GPCCR = 164015 ; CONTROLLER COMMAND REGISTER (WO)
47  GPDIR = 164016 ; DATA IN REGISTER (RO)
48  GPDOR = 164016 ; DATA OUT REGISTER (WO)
49
50  PROM=1 ; SET TO 1 WHEN INTERRUPT VECTORS LOADED INTO PROM;
51  OTHERWISE THEY ARE LOADED DYNAMICALLY BY GPIV1.
52  RAM=0 ; ALSO, NO LOCAL RAM VARIABLES IF PROM=1.

```

```

54 000000
55 000000 010046
56 000000 000005
57 000002 000005
58 000004 105037 164006
59
60 000010 113700 164010
61 000014 032700 000140
62 000020 001401
63 000022 000000
64 000024 042700 177740
65 000030 110037 164010
66
67
68
69
70 000034 012600
71 000036 016537 000002 000000C
72 000044 005037 000000C
73 000050 112737 000240 164000
74 000056 112737 000010 164015
75 000064 000207
76
77
78 000066
79
80 000066 016537 000002 000000C
81 000074 017537 000004 000000C
82 000102 005037 000000C
83 000106 112737 000241 164000
84 000114 000001
85 000116 005737 000000C
86 000122 001004
87 000124 023737 000000C 000000C
88 000132 003370
89
90 000134 112737 000240 164000
91 000142 005037 000000C
92 000146 013775 000000C 000006
93 000154 000207
94
95
96 000156
97
98
99 000156 016537 000002 000000C
100 000164 017537 000004 000000C
101 000172 112737 000340 164000
102 000200 112737 000102 164012
103 000206 000001
104 000210 005737 000000C
105 000214 003374
106 000216 112737 000240 164000
107 000224 105037 164012
108 000230 000207

***** GP1B INITIALIZATION SUBROUTINE *****
GPINI:
MOV R0,-(SP)
;SAVE REGISTER 0 ON STACK
;??NEEDED??
INITIALIZE GP1B AUX REGISTER
;SET HOLD RFD ON EOI
;GET CONTROLLER ADDRESS
;TEST CERTAIN SWITCHES ON CONTROLLER
;OKAY; CONTINUE
;ERROR: SAC AND/OR EXT SET
;ENABLE TALKING AND LISTENING
;LOAD CONTROLLER ADDRESS
;DON'T DO IF VECTOR SPACE IS ROM
;LOAD INTERRUPT VECTOR
;AND PSW
;RESTORE REGISTER 0
;LOAD ADDRESS TO GO IF TRIGGER
;CLEAR EOI FLAG
;ENABLE GET INTERRUPTS
;ENABLE MASTER INTERRUPT
;BUS IS NOW INITIALIZED
RTS

;**** GP1B STRING INPUT SUBROUTINE ****
GPIN:
BISB #100,0#CPACR
MOV 2(R5),0#CPBIN
MOV 04(R5),0#CPMAX
CLR 0#CPNCH
MOV 241,0#CPIMR
WAIT
TST 0#CPEOF
BNE 208
CHP 0#CPMAX,0#CPNCH
BGT 108
MOV 240,0#CPIMR
CLR 0#CPEOF
MOV 0#CPNCH,06(R5)
RTS

;***** GP1B STRING OUTPUT SUBROUTINE *****
GPOUT:
BISB #100,0#CPACR
MOV 2(R5),0#CPBOU
MOV 04(R5),0#CPNCH
MOV 340,0#CPIMR
MOV 102,0#CPSPR
WAIT
TST 0#CPNCH
BGT 108
MOV 240,0#CPIMR
CLR 0#CPSPR
RTS

```

```

110 000232 032737 001000 164000 000000 000000 000000 000000 000000 000000
111 000232 001407 000002 164015 000002 164015 000002 164015 000002 164015
112 000240 142737 000002 164015 000002 164015 000002 164015 000002 164015
113 000240 142737 000002 164015 000002 164015 000002 164015 000002 164015
114 000242 132737 000002 164015 000002 164015 000002 164015 000002 164015
115 000250 132737 000002 164015 000002 164015 000002 164015 000002 164015
116 000256 001371 000002 164015 000002 164015 000002 164015 000002 164015
117 000260 132737 000001 164000 000001 164000 000001 164000 000001 164000
118 000260 132737 000001 164000 000001 164000 000001 164000 000001 164000
119 000266 001013 000001 164000 000001 164000 000001 164000 000001 164000
120 000270 132737 000100 164000 000100 164000 000100 164000 000100 164000
121 000276 001026 000040 164000 000040 164000 000040 164000 000040 164000
122 000300 132737 000001 164000 000001 164000 000001 164000 000001 164000
123 000306 001001 000001 164000 000001 164000 000001 164000 000001 164000
124 000310 000436 000001 164000 000001 164000 000001 164000 000001 164000
125 000312 000177 000000 000000 000000 000000 000000 000000 000000 000000
126 000312 000177 000000 000000 000000 000000 000000 000000 000000 000000
127 000312 000177 000000 000000 000000 000000 000000 000000 000000 000000
128 000316 132737 000002 164000 000002 164000 000002 164000 000002 164000
129 000324 001403 000001 000000 000001 000000 000001 000000 000001 000000
130 000326 012737 000001 000000 000001 000000 000001 000000 000001 000000
131 000326 012737 000001 000000 000001 000000 000001 000000 000001 000000
132 000334 113777 164016 000000 164016 000000 164016 000000 164016 000000
133 000342 005237 000000 000000 000000 000000 000000 000000 000000 000000
134 000346 005237 000000 000000 000000 000000 000000 000000 000000 000000
135 000352 000415 000000 000000 000000 000000 000000 000000 000000 000000
136 000352 000415 000000 000000 000000 000000 000000 000000 000000 000000
137 000354 117737 000000 164016 000000 164016 000000 164016 000000 164016
138 000362 005237 000000 000000 000000 000000 000000 000000 000000 000000
139 000362 005237 000000 000000 000000 000000 000000 000000 000000 000000
140 000366 005337 000000 000000 000000 000000 000000 000000 000000 000000
141 000372 003401 000000 000000 000000 000000 000000 000000 000000 000000
142 000374 000404 000000 000000 000000 000000 000000 000000 000000 000000
143 000376 132737 000000 164000 000000 164000 000000 164000 000000 164000
144 000404 001774 000000 000000 000000 000000 000000 000000 000000 000000
145 000406 000002 000000 000000 000000 000000 000000 000000 000000 000000
146 000406 000002 000000 000000 000000 000000 000000 000000 000000 000000
147 000406 000002 000000 000000 000000 000000 000000 000000 000000 000000
148 000406 000002 000000 000000 000000 000000 000000 000000 000000 000000
149 000406 000002 000000 000000 000000 000000 000000 000000 000000 000000
150 000406 000002 000000 000000 000000 000000 000000 000000 000000 000000
151 000406 000002 000000 000000 000000 000000 000000 000000 000000 000000
152 000406 000002 000000 000000 000000 000000 000000 000000 000000 000000
153 000406 000002 000000 000000 000000 000000 000000 000000 000000 000000
154 000406 000002 000000 000000 000000 000000 000000 000000 000000 000000
155 000406 000002 000000 000000 000000 000000 000000 000000 000000 000000
156 000406 000002 000000 000000 000000 000000 000000 000000 000000 000000

***** GPIB I/O INTERRUPT SERVICE ROUTINE *****

:SRCP::
BIT 1000,GPISR
REQ 200
BITB 2,GPCCR
BITB 2,GPCCR
BNE 100
    "BI" INTERRUPT SET?
    YES
    "BO" INTERRUPT SET?
    YES
    "GET" INTERRUPT SET?
    YES
    INTERRUPT ERROR; TRY AGAIN
    RESTART
    IF DATA RECEIVED FROM BUS
    END-OF-INPUT?
    NO--CONTINUE
    SET EOI FLAG
    GET A CHARACTER
    INCREMENT INPUT ADDRESS
    INCREMENT CHARACTER COUNT
    IF DATA TO SEND TO BUS
    SEND OUT DATA BYTE
    INCREMENT OUTPUT ADDRESS
    DONE WITH STRING?
    YES--CLEANUP
    NO--NEXT?
    DATA-OUT REGISTER CLEAR?
    NO
    ASSEMBLE IF VARIABLES NOT FROM
    E-O-I FOUND? FLAG
    INPUT BUFFER ADDRESS
    OUTPUT BUFFER ADDRESS
    MAXIMUM CHARACTERS ON INPUT
    CHARACTER COUNT FOR INPUT OR OUTPUT
    ADDRESS TO GO IF "GET" RECEIVED

```

(7) MODULE GRMS - This subroutine computes parameters from statistical techniques.

Calling Sequence:

CALL GRMS (N, X, RMS, DC, PEAKMX, PEAKMN)

where the arguments are:

N	Length of sequence
X	Floating-point data sequence
RMS	Returned rms voltage (standard deviation)
DC	Returned dc voltage (average)
PEAKMX	Peak voltage (maximum)
PEAKMN	Peak voltage (minimum)

NOTE: This code was originally written by R.W. Anderson and subsequently modified by R.W. Luckey, both of NRL-USRD.

```

1 .TITLE CRMS
2 .IDENT /02/
3
4 ;SUBROUTINE TO COMPUTE TIME-AVERAGE VOLTAGES: RMS, DC,
5 ;MAXIMUM, AND MINIMUM.
6
7 ;
8 ;AUTHOR: RW ANDERSON (SOMETIME IN 1980)
9 ;MODIFIED: RW LUCKEY (FOR BOTH PEAKS, 1981)
10
11 .FORTRAN CALL--
12 ;CALL CRMS (NPTS,VALS,RMS,AVG,PEAKTX,PEAKMIN)
13
14 ;AC0 - IS THE SUMMING REGISTER
15 ;AC1 - IS THE REGISTER FOR THE SUM OF THE SQUARES
16 ;AC2 -
17 ;AC3 - NUMBER OF POINTS
18 ;AC4 - PEAK MINIMUM - STORAGE - REGISTER NOT GENERALLY USABLE
19 ;AC5 - PEAK MAXIMUM - STORAGE - REGISTER NOT GENERALLY USABLE
20
21 ;R0 - LOOP COUNTER - INITIALIZED TO NPTS
22 ;R1 - ADDRESS OF THE VALUES
23
24 AC0 = %0
25 AC1 = %1
26 AC2 = %2
27 AC3 = %3
28 AC4 = %4
29 AC5 = %5
30
31 CRMS:: SETF      04(R5), AC0
32 LDF      AC0, AC4
33 STF      04(R5), AC0
34 LDF      04(R5), AC5
35 STF      AC0, R0
36 MOV      02(R5), R0
37 LDCIF    R0, AC3
38 MOV      4(R5), R1
39 CLRF     AC0
40 CLRF     AC1
41 LDF      (R1), AC2
42 ADDF     AC2, AC0
43 HULF     AC2, AC1
44 ADDF     AC2, AC1
45 LDF      (R1)+, AC2
46 CHFF     AC5, AC2
47 CFCG
48 BPL      MIN
49 STF      AC2, AC5
50 CHFF     AC4, AC2
51 CFCG
52 BHI      CONT
53 STB      AC2, R0
54 SOB      R0, LOOP
55
56 ; PEAK MINIMUM = X(1)
57 ; STORE IN AC4
58 ; PEAK MAXIMUM = X(1)
59 ; STORE IT IN AC5
60 ; SET UP LOOP COUNTER
61 ; NUMBER OF POINTS (NPTS)
62 ; ADDRESS OF DATA IN REGISTER ONE
63 ; ZERO SUMMING REGISTER
64 ; ZERO SQUARES SUMMING REGISTER
65 ; GET A VALUE
66 ; ADD IT TO SUMMING REGISTER
67 ; SQUARE IT
68 ; ADD IT TO SQUARES SUMMING REGISTER
69 ; GET THE VALUE AGAIN
70 ; SEE IF NEW MAX PEAK
71 ; COPY FUNCTION CODES
72 ; BRANCH IF NOT NEW MAX PEAK
73 ; SAVE NEW MAX PEAK
74 ; SEE IF NEW MIN PEAK
75 ; COPY FUNCTION CODES
76 ; BRANCH IF NOT NEW MIN PEAK
77 ; SAVE NEW MIN PEAK
78 ; LOOP IF NOT FINISHED

```

55	000070	174463	DIVF	AC3,	AC0	AVERAGE (AVG) = SUM / NPTS
56	000072	174975	STF	AC0,	010(R5)	STORE AVERAGE (AVG) - DC
57	000076	173205	LDF	AC5,	AC2	MOVE MAX PEAK INTO AC2
58	000100	173200	SUBF	AC0,	AC2	PEAK MAX = PEAK MAX - AVG (DC)
59	000102	174275	STF	AC2,	012(R5)	STORE PEAK MAX
60	000106	172604	LDF	AC4,	AC2	MOVE MIN PEAK INTO AC2
61	000110	173200	SUBF	AC0,	AC2	PEAK MIN = PEAK MIN - AVG (DC)
62	000112	174275	STF	AC2,	014(R5)	STORE PEAK MIN
63	000116	171000	MULF	AC0,	AC0	SQUARE THE AVERAGE (DC)
64	000120	174563	DIVF	AC3,	AC1	SQUARE AVERAGE = SUNSQ / NPTS
65	000122	173100	SUBF	AC0,	AC1	SUBTRACT AVERAGE SQUARED (DC)**2
66	000124	174100	STF	AC1,	AC0	PUT IT IN AC0
67	000126	004767	JSR	PC,	\$9SORT	GET THE SQUARE ROOT
68	000132	174075	STF	AC0,	06(R5)	STORE RMS
69	000136	000207	RTS	PC,		RETURN FROM SUBROUTINE
70		000001	.END			

- (8) LSIVAR - The following is an alphabetical list (with definitions) of the variables used in the software. All are used by LSIVM except for those used by the subroutines noted in parentheses.

AO	Exponent factor (DGZL)
AD1C	Input port for channel 1
AD2C	Input port for channel 2
AD3C	Input port for channel 3
ADBUF	A/D buffers (three contiguous arrays)
ADS	Numbers of A/D's used
ADST	Measurement status register (A32D)
ADT	DMA to interrupt on (ignore all others)
ADUM	A/D pseudo-register (A32D)
ADWL	Length in bytes of buffer to send to host
AERRS	Automatic retries when A/D errors occur
AVGCOD	Averaging mode (0 = don't, or cycles per averaged subsequence)
BSIZE	Buffer size allowed per channel
BUF1	12-bit integer values from A/D #1
BUF2	12-bit integer values from A/D #2
BUF3	12-bit integer values from A/D #3
CO	Cosine factor (DGZL)
C1	Cosine factor (DGZL)
CFACT	Current factor ($I = CFACT * E$)
CHERRY	Display character buffer
CHNO	Channel number being computed
COMTYP	Computation type (not used)
COV	Covariance (ac power)
DATA	Start of LSIVM input parameter buffer

DCV	Average (dc) voltage from statistical computation
DFD	DFT data array
DIST1	Distortion of channel 1
DIST2	Distortion of channel 2
DIST3	Distortion of channel 3
DIST	Harmonic distortion
DLUN	Display device (1 = terminal, 2 = display)
DSTYLE	Display style (0-5)
ECNT	Counter for A/D error retry
FSAM	Sample frequency (Hz)
FSIG	Signal frequency (Hz)
GAIN1	A/D gain channel 1 (1, 2, 5, 10)
GAIN2	A/D gain channel 2 (1, 2, 5, 10)
GAIN3	A/D gain channel 3 (1, 2, 5, 10)
GAIN	A/D converter gains array
GATED	Gated voltage flag (0 = no, 1 = yes)
GPBIN	Input buffer address (GPIO SR)
GPBOUT	Output buffer address (GPIO SR)
GPEOF	E-O-I found - flag (0 = no, 1 = yes) (GPIO SR)
GPMAX	Maximum characters on input (GPIO SR)
GPNCH	Character count for input or output (GPIO SR)
GTRIG	Address to go to if "GET" received (GPIO SR)
HAR	Number of harmonics to use to compute distortion
HARM	Harmonics - used flag array (DAPD)
HTC	Specific harmonic (line + 1) to compute
I	Line number (DAPD)
IBLEN	Bytes received from IEEE-488 bus

IC	Command byte 1
ICA	Count of A/D conversions
ICE	Count of A/D errors
ICN	Command byte 2 (may be channel number)
IERD	Error status of DFT
INBUF	IEEE-488 bus input packet
IP	Command parameter string
ISW	LSIVM status word
IWD	What DMA to interrupt from (A32D)
LADD	Pointer to displayed parameters
LCTR	Counter for displayed parameters
LDAT	Parameter to display
LPTR	Index for displayed parameters
NCAV	Sequences to average (NCYC/AVGCOD)
NCYC	Cycles per sequence
NCYF	Cycles to compute (AVGCOD); averaging changes
NCYHA	Cycles to compute (for a harmonic)
NSAM	Samples per sequence (DAPD)
PAD	Samples to convert per DMA transfer per channel
PDFT	Phase from DFT
PEKVM	Peak (-) voltage from inspection
PEKVP	Peak (+) voltage from inspection
PHAS1	Phase of channel 1
PHAS2	Phase of channel 2
PHAS3	Phase of channel 3
POWER	Total power (ac and dc)
PSIGN	Phase sign (DAPD)

PTC	Points to compute (PTR/NCYC); averaging changes this
PTR	Samples to transfer to host (samples per sequence)
PWR	Harmonic power (DAPD)
RAA	Argument list for writing buffer to host
RAM	Start of program RAM space
RAN	Address of buffer to send to host
RBUF	One voltage buffer (converted to floating-point)
RMD	Statistical data array
RMSV1	RMS voltage in channel 1
RMSV2	RMS voltage in channel 2
RMSV3	RMS voltage in channel 3
RMSV	RMS voltage from statistical analysis
RPHAS	Phase difference between channels 1 and 2
SO	Sine factor (DGZL)
SJM	Second joint moment (total power)
STACK	Start (top) of program stack
STPT	Number of first sample to transfer to host
STRING	Used by print
T1	DMA #1 pseudo-register (A32D)
T2	DMA #2 pseudo-register (A32D)
T3	DMA #3 pseudo-register (A32D)
VDFT	Total rms voltage from DFT
WAD	What A/D converters used (A32D)
WAV	Average window magnitude
WIND	Window buffer (floating-point)
WMODE	Window mode (0 = no, 1 = yes)
XIMAG	Imaginary voltage part (DAPD)

XREAL

Real voltage part (DAPD,

```

1 .TITLE LSIVAR
2 .IDENT /06/
3
4 GLOBAL VARIABLES (DYNAMIC) FOR LSIVM
5
6
7 .PSECT $VARL,REL,RO,NOT REALLY RO!! TO FOOL TASKBUILDER!!
8 .BLKB 2452 !SHIFT BEGINNING OF RAM TO 4K BOUNDARY
9
10 !BUFFER SIZE ALLOWED PER CHANNEL
11
12 !START OF PROGRAM RAM SPACE
13 !START ('TOP') OF PROGRAM STACK
14
15 !COUNT OF A/D CONVERSIONS
16 !COUNT OF A/D ERRORS
17
18 !IEEE-488 BUS INPUT PACKET
19 !COMMAND BYTE 1
20 !COMMAND BYTE 2 (MAYBE CHANNEL NUMBER)
21 !COMMAND PARAMETER STRING
22 !LSIVM STATUS WORD
23 !BYTES RECEIVED FROM IEEE-488 BUS
24 !CHANNEL NUMBER BEING COMPUTED
25
26 !E-0-1 FOUND? FLAG (0=NO, 1=YES) (GPIOSR)
27 !INPUT BUFFER ADDRESS (GPIOSR)
28 !OUTPUT BUFFER ADDRESS (GPIOSR)
29 !MAXIMUM CHARACTERS ON INPUT (GPIOSR)
30 !CHARACTER COUNT FOR INPUT OR OUTPUT (GPIOSR)
31 !ADDRESS TO GO TO IF "GET" RECEIVED (GPIOSR)
32
33 !START OF LSIVM INPUT PARAMETER BUFFER
34 !START OF A/D CONVERTER GAINS ARRAY
35 !A/D GAIN CHANNEL 1 (1,2,5,10)
36 !A/D GAIN CHANNEL 2 (1,2,5,10)
37 !A/D GAIN CHANNEL 3 (1,2,5,10)
38 !SAMPLES TO TRANSFER TO HOST (SAMPLES PER SEQUENCE)
39 !NUMBER OF FIRST SAMPLE TO TRANSFER TO HOST
40 !NUMBERS OF A/D'S USED (1=1,2=2,4=4,7=ALL)
41 !DMA TO INTERRUPT ON (IGNORE ALL OTHERS)
42 !SAMPLES TO CONVERT PER DMA TRANSFER PER CHANNEL
43 !CYCLES PER SEQUENCE
44 !NUMBER OF HARMONICS TO USE TO COMPUTE DISTORTION
45 !AVERAGING MODE (0=DON'T, OR CYCLES PER AVERAGE SUBSEQ..)
46 !DISPLAY STYLE (0-5)
47 !COMPUTATION TYPE (NOT USED)
48 !AUTOMATIC RETRIES WHEN A/D ERRORS OCCUR
49 !GATED VOLTAGE FLAG (0=NO, 1=YES)
50 !WINDOW MODE (0=NO, 1=YES)
51 !DISPLAY DEVICE (1=TERMINAL, 2=DISPLAY)
52 !SPECIFIC HARMONIC (LINE+1) TO COMPUTE
53 !INPUT PORT FOR CHANNEL 1
54 !INPUT PORT FOR CHANNEL 2
55 !INPUT PORT FOR CHANNEL 3

```

```

57 003676 000000 000000 000000 000000 000000 000000 000000 000000 000000
58 003702 000000 000000 000000 000000 000000 000000 000000 000000 000000
59
60 003706 000000 000000 000000 000000 000000 000000 000000 000000 000000
61 003710 000000 000000 000000 000000 000000 000000 000000 000000 000000
62 003712 000000 000000 000000 000000 000000 000000 000000 000000 000000
63 003714 000000 000000 000000 000000 000000 000000 000000 000000 000000
64 003716 000000 000000 000000 000000 000000 000000 000000 000000 000000
65 003720 000000 000000 000000 000000 000000 000000 000000 000000 000000
66
67 003722 000000 000000 000000 000000 000000 000000 000000 000000 000000
68 003726 000000 000000 000000 000000 000000 000000 000000 000000 000000
69 003728 000000 000000 000000 000000 000000 000000 000000 000000 000000
70 003732 000000 000000 000000 000000 000000 000000 000000 000000 000000
71
72 003736 000000 000000 000000 000000 000000 000000 000000 000000 000000
73 003738 000000 000000 000000 000000 000000 000000 000000 000000 000000
74 003742 000000 000000 000000 000000 000000 000000 000000 000000 000000
75 003746 000000 000000 000000 000000 000000 000000 000000 000000 000000
76 003752 000000 000000 000000 000000 000000 000000 000000 000000 000000
77
78 003756 000000 000000 000000 000000 000000 000000 000000 000000 000000
79 003762 000000 000000 000000 000000 000000 000000 000000 000000 000000
80
81 003766 000000 000000 000000 000000 000000 000000 000000 000000 000000
82 003772 000000 000000 000000 000000 000000 000000 000000 000000 000000
83
84 003776 000000 000000 000000 000000 000000 000000 000000 000000 000000
85 004004 000000 000000 000000 000000 000000 000000 000000 000000 000000
86
87 004006 000000 000000 000000 000000 000000 000000 000000 000000 000000
88 004010 000000 000000 000000 000000 000000 000000 000000 000000 000000
89 004012 000000 000000 000000 000000 000000 000000 000000 000000 000000
90 004014 000000 000000 000000 000000 000000 000000 000000 000000 000000
91 004016 000000 000000 000000 000000 000000 000000 000000 000000 000000
92 004020 000000 000000 000000 000000 000000 000000 000000 000000 000000
93 004022 000000 000000 000000 000000 000000 000000 000000 000000 000000
94 004024 000000 000000 000000 000000 000000 000000 000000 000000 000000
95 004026 000000 000000 000000 000000 000000 000000 000000 000000 000000
96 004030 000000 000000 000000 000000 000000 000000 000000 000000 000000
97
98 004032 000000 000000 000000 000000 000000 000000 000000 000000 000000
99 004034 000000 000000 000000 000000 000000 000000 000000 000000 000000
100 004036 000000 000000 000000 000000 000000 000000 000000 000000 000000
101 004040 000000 000000 000000 000000 000000 000000 000000 000000 000000
102 004042 000000 000000 000000 000000 000000 000000 000000 000000 000000
103

```

; SIGNAL FREQUENCY (HZ)
; SAMPLE FREQUENCY (HZ)
; POINTS TO COMPUTE (PTR/NCYC); AVERAGING CHANGES
; CYCLES TO COMPUTE (AVGCCD); AVERAGING CHANGES
; CYCLES TO COMPUTE (FOR A HARMONIC)
; SEQUENCES TO AVERAGE (NCYC/AVGCCD)
; COUNTER FOR A/D ERROR MATH
; LENGTH IN BYTES OF BUFFER TO SEND TO HOST
; START OF DFT DATA ARRAY
; TOTAL RMS VOLTAGE FROM DFT
; PHASE FROM DFT
; HARMONIC DISTORTION
; START OF TIME-INTEGRATION DATA ARRAY
; RMS VOLTAGE FROM TIME INTEGRATION
; PEAK(+) VOLTAGE FROM INSPECTION
; PEAK(-) VOLTAGE FROM INSPECTION
; AVERAGE (DC) VOLTAGE FROM TIME INTEGRATION
; TOTAL POWER (SUM OF E*1)
; CURRENT FACTOR (I=CFACT*E)
; SECOND JOINT MOMENT
; COVARIANCE
; ARGUMENT LIST FOR WRITING BUFFER TO HOST
; ADDRESS OF BUFFER TO SEND TO HOST
; RAM VARIABLES USED BY A32D
; WHAT A/D CONVERTERS USED (A32D)
; WHAT DMA TO INTERRUPT FROM (A32D)
; DMA #1 PSEUDO-REGISTER (A32D)
; DMA #2 PSEUDO-REGISTER (A32D)
; DMA #3 PSEUDO-REGISTER (A32D)
; A/D PSEUDO REGISTER (A32D)
; MEASUREMENT STATUS REGISTER (A32D)
; CHANNEL 1 PORT ADDRESS * 400
; CHANNEL 2 PORT ADDRESS * 400
; CHANNEL 3 PORT ADDRESS * 400
; COUNTER FOR DISPLAYED PARAMETERS
; INDEX FOR DISPLAYED PARAMETERS
; POINTER TO DISPLAYED PARAMETERS
; PARAMETER TO DISPLAY
; USED BY PRINT

```

105
106 004056 000000 000000 XREAL:: .FLT2 0.0
107 004062 000000 000000 XIMAG:: .FLT2 0.0
108 004066 000000 000000 PHR:: .FLT2 0.0
109 004072 000000 000000 PSIGN:: .WORD 0
110 004074 000000 000000 L:: .WORD 0
111 004076 000000 000000 NSAM:: .WORD 0
112 004100 000000 000000 HARM:: .BLKW 20
113
114 004150 000000 000000 IEND:: .WORD 0

```

```

;RAM VARIABLES USED BY DAPD
;REAL VOLTAGE PART (DAPD)
;IMAGINARY VOLTAGE PART (DAPD)
;HARMONIC POWER (DAPD)
;PHASE SIGN (DAPD)
;LINE NUMBER (DAPD)
;SAMPLES PER SEQUENCE (DAPD)
;HARMONICS-USED FLAG ARRAY (DAPD)
;ERROR STATUS OF DFT

```

```

116
117 004152 000000 000000 A0:: .FLT4 0.0
118 004160 000000 000000 C0:: .FLT4 0.0
119 004170 000000 000000 C1:: .FLT4 0.0
120 004200 000000 000000 S0:: .FLT4 0.0
121 004210 000000 000000

```

```

;RAM VARIABLES USED BY DCZL
;FACTOR (DCZL)
;COSINE FACTOR (DCZL)
;COSINE FACTOR (DCZL)
;SINE FACTOR (DCZL)

```

```

121 004312 000000 000000 CHERRY:: .BLKB 36
122 004326 000000 000000 RNSV1:: .FLT2 0.0
123 004356 000000 000000 RNSV2:: .FLT2 0.0
124 004262 000000 000000 RNSV3:: .FLT2 0.0
125 004266 000000 000000 PHAS1:: .FLT2 0.0
126 004272 000000 000000 PHAS2:: .FLT2 0.0
127 004276 000000 000000 PHAS3:: .FLT2 0.0
128 004302 000000 000000 DIST1:: .FLT2 0.0
129 004306 000000 000000 DIST2:: .FLT2 0.0
130 004312 000000 000000 DIST3:: .FLT2 0.0
131 004316 000000 000000 RPHAS:: .FLT2 0.0
132 004322 000000 000000
133 004326 000000 000000 ADBUF::
134 004326 000000 000000 BU1:: .BLKW BSIZE
135 004326 000000 000000 BU2:: .BLKW BSIZE
136 010326 000000 000000 BU3:: .BLKW BSIZE
137 014326 000000 000000
138 020326 000000 000000 RBUF:: .BLKW 2*BSIZE
139 020326 000000 000000 WAV:: .FLT2 0.0
140 030326 000000 000000 WIND:: .BLKW 2*BSIZE
141 030332 000000 000000
142
143 000001 .END

```

```

;CHERRY DISPLAY CHARACTER BUFFER
;RMS VOLTAGE IN CHANNEL 1
;RMS VOLTAGE IN CHANNEL 2
;RMS VOLTAGE IN CHANNEL 3
;PHASE OF CHANNEL 1
;PHASE OF CHANNEL 2
;PHASE OF CHANNEL 3
;DISTORTION OF CHANNEL 1
;DISTORTION OF CHANNEL 2
;DISTORTION OF CHANNEL 3
;PHASE DIFFERENCE BETWEEN CHANNELS 1 AND 2
;A/D BUFFERS (THREE CONTIGUOUS ARRAYS)
;12-BIT INTEGER VALUES FROM A/D #1
;12-BIT INTEGER VALUES FROM A/D #2
;12-BIT INTEGER VALUES FROM A/D #3
;ONE VOLTAGE BUFFER (CONVERTED TO FLOATING-POINT)
;AVERAGE WINDOW MAGNITUDE
;WINDOW BUFFER (FLOATING-POINT)

```

(BLANK PAGE)

APPENDIX F

Voltmeter Memory Map

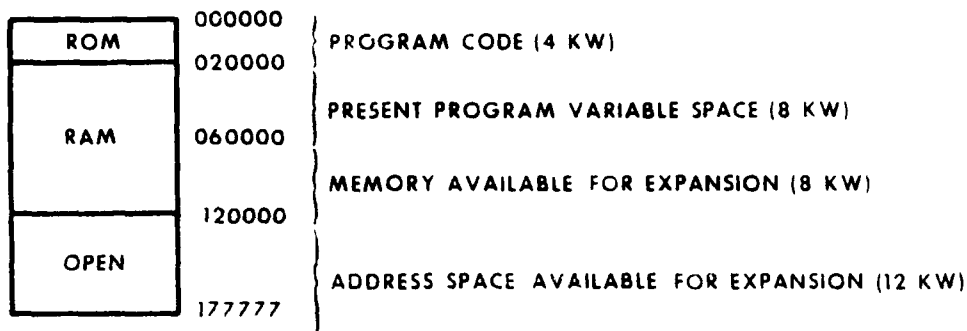


Fig. F1 - Voltmeter memory map.

The ROM code used by this version of the voltmeter occupies approximately 15300 (octal) bytes of memory, so there is room for expansion.

The program variable space allows, at present, data buffers of 1024 samples, but this can be expanded by modifying the parameter BSIZE in LSIVAR when the program is rebuilt. Note that for each sample a buffer is expanded, memory is used as follows:

- 6 bytes for the sampled A/D waveform (2 bytes per channel),
- 4 bytes for the integer to floating-point conversion buffer,
- 4 bytes for the floating-point data window.

This requires approximately 7K words of RAM memory for each additional 1024 samples.